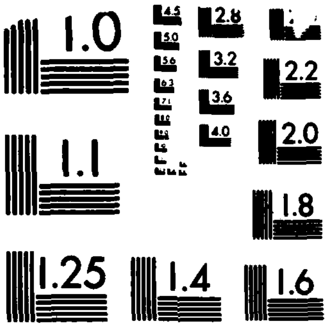
AD-A138 587 INTEGRATED TESTING AND MAINTENANCE TECHNOLOGIES(U) 1/4
BOEING AEROSPACE CO SEATTLE WA R O DENNEY ET AL.
DEC 83 AFWAL-TR-83-1183 F33615-81-C-1517

UNCLASSIFIED F/G 5/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

AFWAL-TR-83-1183

INTEGRATED TESTING AND MAINTENANCE TECHNOLOGIES

Robert O. Denney
Mike J. Partridge
Roger B. Williams

Boeing Aerospace Company
P.O. Box 3999
Seattle, Washington 98124

December 1983

Final Technical Report for Period 25 September 1981 - 15 September 1983

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Chahira M. Hopper

CHAHIRA M. HOPPER
Project Engineer
System Integration Branch
System Avionics Division

Terrance A. Brim

TERRANCE A. BRIM, Chief
System Integration Branch
System Avionics Division
Avionics Laboratory

FOR THE COMMANDER

Raymond D Bellem

RAYMOND D. BELLEM, LT COL, USAF
Deputy Chief
System Avionics Division
Avionics Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/AAAS-2 *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFWAL-TR-83-1183	2. GOVT ACCESSION NO. AD-A138587	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) INTEGRATED TESTING AND MAINTENANCE TECHNOLOGIES		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 25 Sept. 1981 - 15 Sept.1983
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Robert O. Denney Mike J. Partridge Roger B. Williams		8. CONTRACT OR GRANT NUMBER(s) F33615-81-C-1517
9. PERFORMING ORGANIZATION NAME AND ADDRESS Boeing Aerospace Company P. O. Box 3999 Seattle, Washington		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 20030919
11. CONTROLLING OFFICE NAME AND ADDRESS Avionics Laboratory (AFWAL/AAAS-2) Air Force Wright Aeronautical Laboratories Air Force Systems Command, USAF Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December 1983
		13. NUMBER OF PAGES 320
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence	Fault Isolation
Avionics Testing	Integrated Testing and Maintenance
Built-in Test (BIT)	Maintenance
Expert Systems	Onboard Testing
Fault Detection	System Integrated Test

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Maintenance of weapon systems is becoming an increasingly important consideration in weapon system development, because the cost of maintenance is a significant portion of the life cycle cost of the system. The objective of the Integrated Testing and Maintenance Technologies effort is to define requirements for an onboard test system for the avionic suite planned for tactical fighters in the 1990's. Problems with current onboard test systems were analyzed to determine where improvements could be made. In addition,

Block 20 (cont'd)

the anticipated avionic architecture and mission of the 1990's were evaluated to determine the impact on maintenance capability. Requirements for the Integrated Testing and Maintenance System were developed and documented in a system specification. Identified improvements over current systems include better filtering of intermittent failure reports, better isolation of intermittent failures through the use of recorded data, more extensive use of system-level tests of mission operational data and a man-machine interface providing more information to the maintenance technician. In addition, artificial intelligence applications were evaluated to determine where they might be effectively applied to ITM. A design concept for a fault classification expert system was developed.

PREFACE

This report documents the results of the effort in accordance with Air Force Contract F33615-81-C-1517, Integrated Testing and Maintenance Technologies. An interim product of the effort is an ITM system specification. The ITM system specification is a companion document to this report.

This contract and an associated contract (F33615-81-C-1520, Multibus Avionics Architecture Design Study, awarded to TRW, Defense and Space Systems Group) provide input to the Advanced System Integration Demonstrations (ASID) program, designated as PAVE PILLAR. The ASID program goal is to define, develop, and evaluate new approaches to integrated avionic system technology to improve availability, operational effectiveness, and survivability of tactical fighter aircraft.

The work was accomplished by the Engineering Technology Organization of Boeing Aerospace Company under the management of T. A. Nicolino.

The contract was administered by the Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command. Technical and administrative support was provided initially by Capt. Daniel D. Quaderer and later transferred to Chahira M. Hopper, with interim support from David A. Zann. Management and additional technical support was provided by Diane Summers and subsequently by Mark Thullen (AFWAL/AAAS-2).

TABLE OF CONTENTS

SECTION PAGE

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

1.0 INTRODUCTION

1.1 PROBLEM

Maintenance of weapon systems is becoming an increasingly important consideration in weapon system development. Improvement in the maintenance capability of a weapon system greatly reduces total cost of the system because the cost of maintenance is a significant portion of life cycle cost. Another important cost factor is that improved maintenance increases availability, which reduces the number of systems that need to be acquired and hence lowers the acquisition cost of the weapon system.

Difficulty in maintaining avionic systems is increasing due to the growing complexity of avionics and decreasing skill of maintenance personnel. In the past, test capability at the system level and at the subsystem level had been designed after the operational design was complete, resulting in less than optimal test capability. Experience by the users of the avionics systems indicates that the onboard test systems lack sufficient unambiguous fault detection and fault isolation. This had caused an unacceptable rate of failure reports during a mission that cannot be duplicated during maintenance and an unacceptable rate of components returned for repair that retest OK.

1.2 OBJECTIVE

The objective of the Integrated Testing and Maintenance (ITM) Technologies study is to define the requirements for an onboard test system for the avionics suite planned for tactical fighters in the 1990's. The avionics suite and architecture to be used to develop ITM has been defined as a product of an associated contract, the Multibus Avionic Architecture Design Study (MAADS), contract number F33615-81-C-1520. The ITM effort is to develop the onboard test capability using existing avionic system resources to the maximum extent possible.

1.3 SCOPE

The scope of the ITM effort is the development of test and maintenance requirements for all aspects of organizational-level maintenance of the airborne avionics. This includes preflight checkout, inflight monitoring, and postflight corrective maintenance. The effort does not address the requirements of intermediate-level or depot-level maintenance except to the extent that organizational-level requirements are derived from maintenance activity at the intermediate level or depot level. An example is the requirement to record selected data in flight to be used in intermediate- or depot-level testing.

ITM design has been oriented toward implementing a two-level maintenance structure—organizational and depot levels. However, the current design does not preclude the adaption of a three-level maintenance structure—organizational, intermediate, and depot levels.

ITM addresses the test requirements of all airborne avionics, but excludes nonavionic test requirements of, for example, engines and airframe. Included in the avionics, however, are systems not traditionally included such as the flight/propulsion control system, the electrical power management system, and the stores management system. The test requirements of the nonavionic systems are considered to the extent that provisions are made within the ITM system for central collection of the nonavionic test data.

The requirements for an ITM system are documented in the ITM system specification, and this report documents the development of those requirements. Explanation, rationale, and analyses are provided in this report for those requirements that represent changes or new approaches to onboard testing of tactical aircraft avionics.

1.4 FUNCTIONAL OVERVIEW OF ITM

To help provide an understanding of the material presented in this report, this section provides a description of the ITM system as defined by the requirements developed as a result of the study.

ITM provides for a complete, self-contained capability for integrated organizational-level avionics testing and maintenance activity. This involves testing at the flight line for preflight checkout and postflight diagnosis as well as for in-flight monitoring and recording of system status and test data. In addition to the organizational-level maintenance support, ITM provides for recorded data to support intermediate-level (if included in maintenance concept) and depot-level maintenance activity. ITM uses mission resources that are provided for the other mission functions, that is, mission processors, multiplex buses, controls and displays, system mass memory, and a data transfer unit. Hence, ITM is implemented primarily as a software function. The software is provided as two separate end items—the Operational Test Program (OTP), which provides the capability to support preflight and postflight testing and the Inflight Test Program (IFTP), which provides the capabilities that support inflight testing. The Maintenance Interface Unit (MIU) provides the only ITM-unique hardware interface and is used for postflight maintenance.

ITM establishes preflight operational readiness through detecting failures, supporting reconfiguration, and providing system status to the pilot and maintenance crew. ITM assesses and maintains system capability in flight through detecting failures, supporting reconfiguration, and providing system status to the pilot. ITM supports postflight corrective maintenance by isolating failures to a line replaceable unit (LRU) and verifying repair. In addition, ITM provides support to intermediate (if required) and depot shop operations through recorded data to help isolate failures within LRU's and for maintenance history and trend analysis use.

ITM functional areas consist of the OTP that contains separate preflight and postflight modes, the IFTP that is to be included as part of the Operational Flight Program (OFP), and the MIU that provides an interface to the ground crew during remove or replace and service activity. The major information flow paths between these functional areas are shown in figure 1.

The ITM system can be applied to various avionics configurations and missions. A specific avionics configuration is established for each weapon system ITM is used on. ITM is imbedded in the weapons system avionics processing function: in the mission or core processing element in systems with a centralized or master control form of architecture, or in any nodal processing element having complete system access in a distributed control form of architecture.

The ITM function is independent of specific weapon system missions. Preflight and postflight operations of the OTP occur outside mission timelines. Inflight ITM functions do not adversely impact weapon system performance requirements or pilot and weapon system performance. Where ITM shares resources with avionics functions (e.g., processors, displays, storage, data buses), it is designed for minimum load and minimum

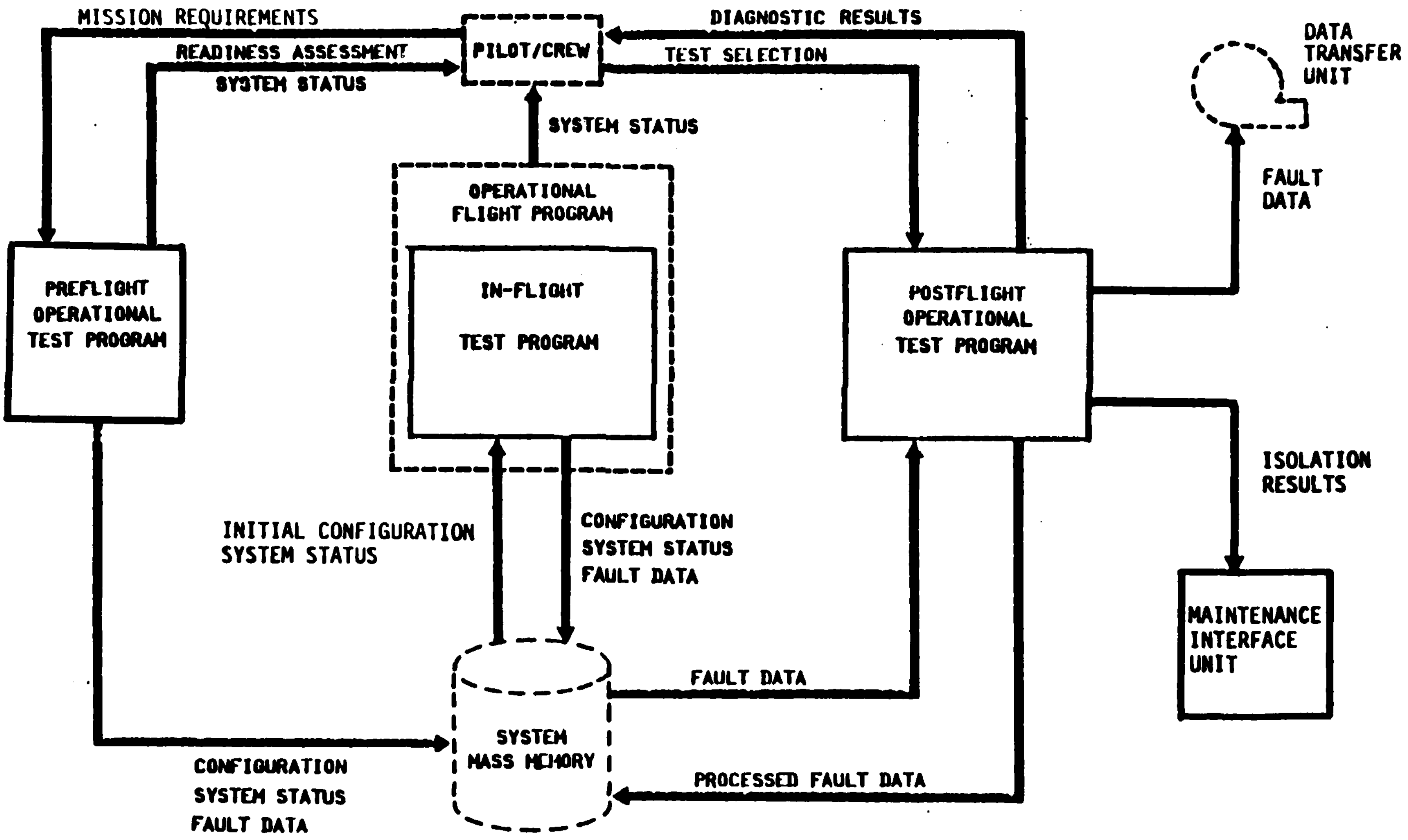

MISSION REQUIREMENTS
PILOT/CREW
DIAGNOSTIC RESULTS
READINESS ASSESSMENT
SYSTEM STATUS
TEST SELECTION
DATA TRANSFER UNIT
SYSTEM STATUS
OPERATIONAL FLIGHT PROGRAM
FAULT DATA
PREFLIGHT OPERATIONAL TEST PROGRAM
IN-FLIGHT TEST PROGRAM
POSTFLIGHT OPERATIONAL TEST PROGRAM
ISOLATION RESULTS
INITIAL CONFIGURATION SYSTEM STATUS
CONFIGURATION SYSTEM STATUS FAULT DATA
MAINTENANCE INTERFACE UNIT
FAULT DATA
SYSTEM MASS MEMORY
CONFIGURATION SYSTEM STATUS FAULT DATA
PROCESSED FAULT DATA

Figure 1. ITM System Functional Area Diagram

interference with inflight avionics functions. Noncritical alarms and displays will be inhibited during critical mission phases.

The ITM system-level block diagram is shown in figure 2 for a centralized control architecture avionics system. Only the OTP and IFTP software, resident in the mission computer, and the MIU are dedicated ITM functional blocks. The other blocks in the diagram represent system elements providing ITM support functions.

The OTP resides with the OFP on the system mass memory, and the IFTP resides on the system mass memory as part of the OFP. The specific ITM program and mode are selectable for loading by the pilot or ground maintenance crew (as applicable). The ground maintenance crew has the capability to interact with the ITM system to specify mission equipment requirements in the preflight mode and test sequences and diagnostic procedures in the postflight mode. Testing is initiated automatically by the ITM system and the results are displayed and/or recorded as appropriate. ITM provides the option for pilot or maintenance crew to select a specific test or tests to be run if a complete test sequence is not necessary.

Built-in test (BIT) data from each LRU in each subsystem provides the primary ITM input. ITM acquires this information over the system multiplex buses. ITM has the capability to acquire BIT information in two distinct ways:

a. Obtain BIT data on an initiated basis during which ITM interrupts normal processing to acquire the BIT data. This is used primarily in preflight and postflight operation, but ITM has the capability to request any specific BIT information in flight (automatically or with pilot intervention) as an aid to diagnosis of critical problems. The weapon system OFP is responsible for safeguards and time-sequencing of the ITM requests to ensure flight safety and maximize mission success.

b. Obtain BIT data on a continuous (where applicable) and on an interleaved basis in a noninterruptive manner (i.e., not affect normal operation).

ITM has the capability to acquire mission data for reasonableness tests to supplement continuous and interleaved BIT data and perform statistical tests to detect degraded performance and develop trending information.

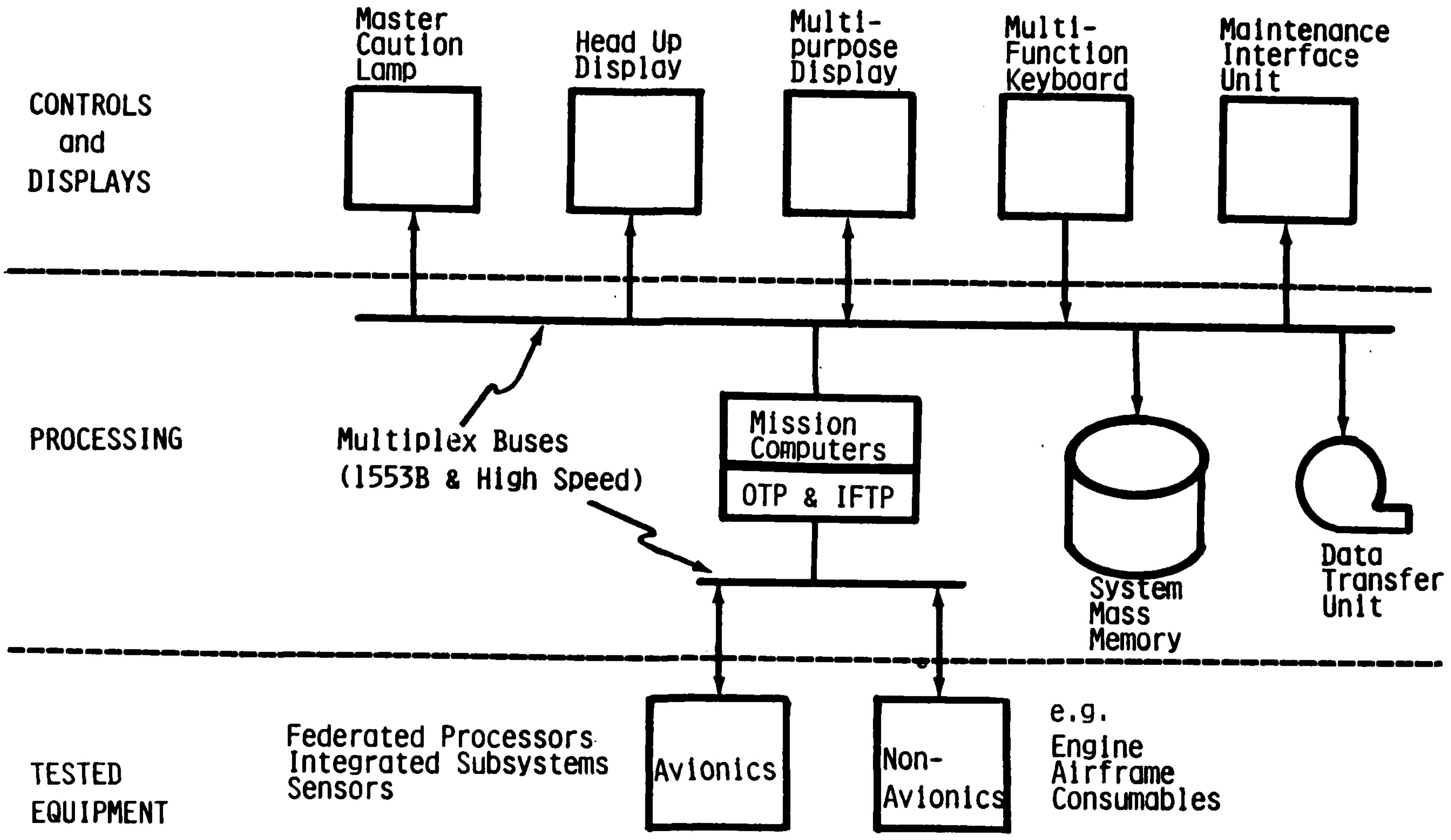


Figure 2. ITM System-Level Block Diagram

The pilot interface with ITM is minimized for normal inflight operations. An interactive pilot interface capability is provided to allow participation (as desired by the pilot) in checkout and diagnostic decisions or to monitor the approach and results of OTP or IFTP action. All cockpit interfaces with ITM are through the controls and displays provided as part of the weapon system.

ITM incorporates provisions to respond to abnormal weapon system operation. ITM is able to abort preflight OTP operation at any time to allow quick takeoff. ITM has the capability to abort preflight OTP test sequences at any time and switch to the postflight OTP mode to allow LRU isolation. ITM provides capability to abort the postflight OTP and switch to the preflight OTP to allow for quick takeoff. ITM provides the capability to stay in the IFTP for continuous monitoring on the ground when through-flight (quick turnaround) is required.

ITM has the following additional features:

a. The ability to set and use test tolerances (setable only under appropriate configuration control methods for the ITM data base).

b. The ability to retest failed LRU's and, if operating correctly, restore them to a functional status (primarily for failures because of environmental causes).

c. The ability to acquire and record environmental data and time for the system.

d. The ability to integrate test results from BIT tests, reasonableness tests, system-level tests, and environmental inputs to allow a thorough isolation and analysis capability of intermittent faults in the postflight OTP.

e. The ability to provide intermittent fault "thresholding" (i.e., number, frequency, and duration threshold) to filter nuisance alarms from the pilot.

f. The ability to record and offload time-sequenced data (e.g., environmental, detected faults, fault-isolation data, reconfiguration action taken, and relevant ancilliary data such as equipment operating modes at time of failure, system and subsystem modes) for postflight ITM analysis and diagnostics, offline analysis, and historical information on specific subsystems and LRU's.

1.5 COST EFFECTIVENESS

In developing requirements for an ITM system, it is necessary to examine the cost effectiveness of the proposed capabilities. This becomes a significant task because ITM affects all major contributions to the cost effecti eness of a system.

Figure 3 shows the relationship of various contributors to the cost effectiveness of a system. Indicated is the general impact of ITM on each of the particular factors. Development of ITM negatively affects acquisition cost due to higher design and development costs and acquisition of added computing capability and BIT hardware. Operation costs are decreased due to lower maintenance costs. Capability is increased because the pilot, with an accurate assessment of the status of the system, can make better inflight operating decisions. The inherent reliability (failure rate of the components of the system) is decreased because of additional hardware associated with the added computing capability and BIT hardware. Operational reliability (failure rate of the system) is increased because of enhanced fault tolerance capability. Finally, maintainability is increased because less time and fewer resources are required for maintenance.

Detailed analysis of these factors is not possible without complete definition of the system including fleet size, acquisition cost, basing concept, operating concept, maintenance concept, and equipment complement.

1.6 ARTIFICIAL INTELLIGENCE APPLICATIONS TO ITM

After developing the requirements for ITM, the various fields of Artificial Intelligence were examined to determine which could be effectively applied in implementing selected functions of ITM. The study concluded that expert systems could be used, and an implementation concept was developed for an expert system to perform the fault classification task at the start of postflight processing.

1.7 REPORT ORGANIZATION

This report is organized to provide an orderly presentation of the results of the contracted effort. The order and content of the various sections was chosen to provide background early in the report for material presented later. There is no specific

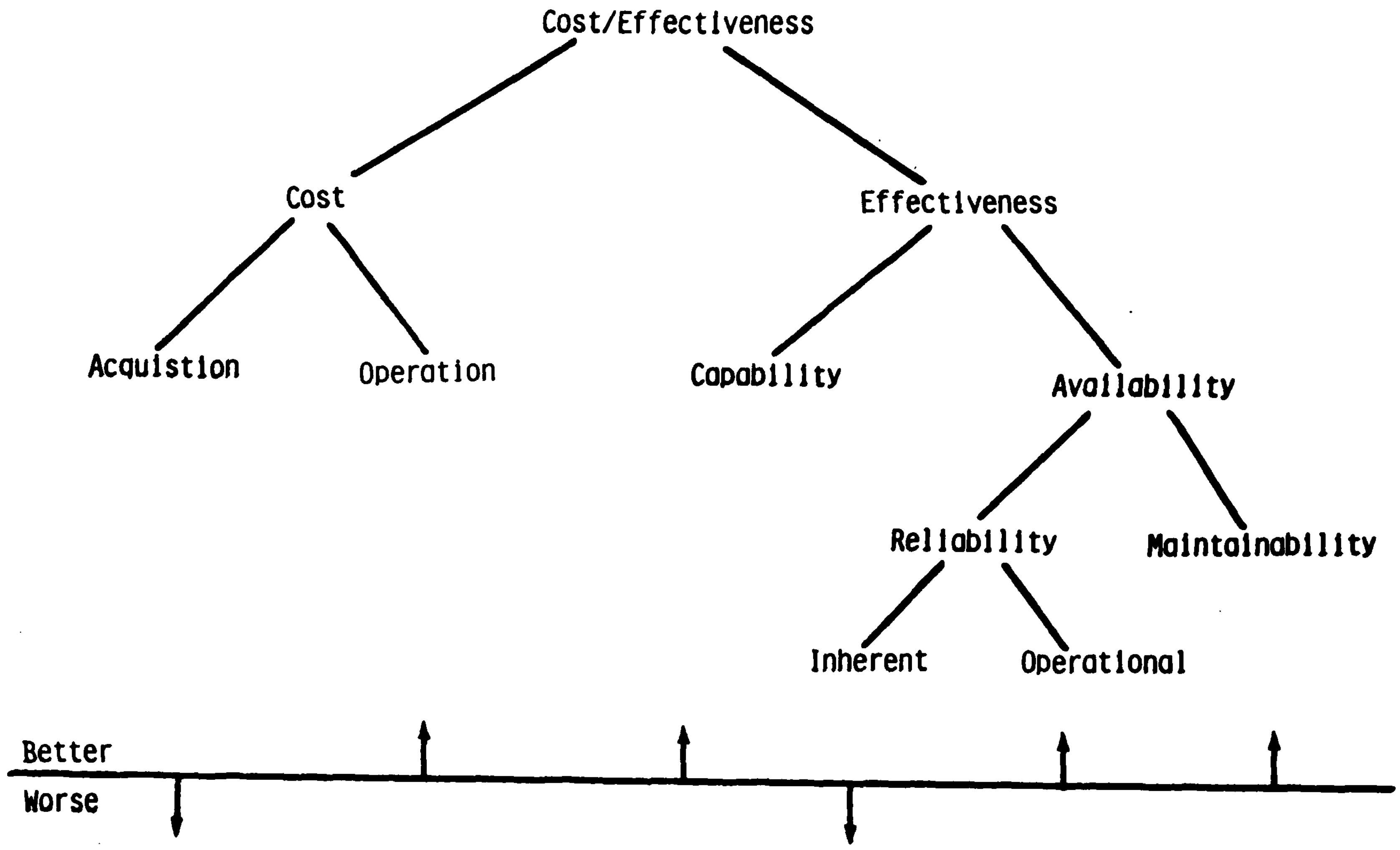


Figure 3. ITM Impact On System Cost/Effectiveness

relationship between the report sections and the various tasks of the contract statement of work.

Section 2.0 defines the problems currently encountered with the test and maintenance of avionics and analyzes the various causes of these problems. Section 3.0 provides background information about various aspects of onboard testing of the F-15, F-16, and F-18 and analyzes their specified performance. Section 4.0 introduces the initial effort to develop ITM system requirements by defining what capabilities should be provided by ITM. Section 5.0 defines the avionics architecture (both functions and structure) for which the ITM requirements were developed. Section 6.0 explains and provides rationale for the requirements documented in the ITM system specification. Many of the detailed requirements are not presented in this report, necessitating reference to the ITM system specification. Section 7.0 presents a cost analysis for the impact on system life cycle cost as a result of implementing the ITM requirements. The investigation into artificial intelligence applications to ITM and the implementation concept for the fault classification expert system are presented in section 8.0.

Appendix A documents an analysis of the impact of incorporating ITM capability into the AFWAL laboratory implementation of the Digital Avionics Information System. Appendixes B through E contain information for which the content or format was inappropriate for inclusion in the main body of the text. These are referenced as appropriate in the report.

2.0 PROBLEM DEFINITION

2.1 APPROACH

There is a consensus that problems exist with testing and maintenance of fielded avionic systems. There is not, however, complete agreement on the extent and cause of the problems or solutions to them.

Our approach was to acquire information from many different sources to determine the extent and causes of the problem and then proceed to develop an approach to solving it. Sources of information included published articles and technical reports, unpublished presentations, discussions with contractors, and unpublished reports of discussions with maintenance personnel.

2.2 PROBLEMS

Review of the above sources yielded five primary problems with current onboard test and maintenance systems. These are:

a. Failure indications that occur during operation but cannot be duplicated (CND) during maintenance.

b. Line replaceable units that are removed from systems during maintenance but retest OK (RTOK) during intermediate-level or depot-level testing.

c. Maintenance personnel have little confidence in the capability and reliability of the automatic test systems.

d. Test and maintenance of avionics requires increasing time and technicians with greater skill levels.

e. When the automatic test systems fail to properly isolate failures, the maintenance personnel have few or no resources to help resolve the problem.

These five test and maintenance problems are interrelated. For example, RTOK's can result from CND's when good units are replaced on the flight line and sent to the shop. This occurs when a failure indication cannot be duplicated, but there is a de facto pressure for some corrective maintenance to occur. The third and fourth problems are partially the result of the other three.

Of these problems, the first two represent major contributors to the cost of maintenance. The third is more psychological than directly cost related. The last two are secondary contributors to maintenance cost.

2.3 CAUSES

The causes of the above problems were identified from the same sources of information. These are listed in the matrix in figure 4, which associates each of the problems with its various causes. The following is a description of each of the problem causes.

Mode of operation dependency—Mode of operation dependency reflects a built-in test (BIT) technique in which functions can only be tested when they are being used. Therefore, when a function is used only in a particular mode of operation, a failure in that function will only be detected by BIT when the equipment is in that particular mode of operation. The effect is that failure indications may not be duplicated unless the conditions are duplicated.

Environmental dependency—Environmental conditions can cause failures to occur. A failure caused by sensitivity to environmental conditions may occur; but when conditions change the failure indication and the failure may disappear.

False alarms—False alarms cause unnecessary maintenance. There is little agreement as to what constitutes a false alarm. A possible resolution to this problem will be discussed in more detail in section 2.4.

Intermittent faults—These are failures that occur several or more times intervened with proper operation. Like mode-dependent and environmental-dependent failure indications, these may appear and disappear as conditions change.

	CAUSES								
PROBLEM	MODE OF OPERATION DEPENDENCY	ENVIRONMENT DEPENDENCY	FALSE ALARMS	INTERMITTENT FAULTS	INADEQUATE FAULT ISOLATION	INCOMPATIBILITY OF TESTS AND TEST TOLERANCES	OUT OF SPEC. BUT SYSTEM STILL OPERATES	FAULTS IN BIT HARDWARE	TEST DATA NOT ACCESSIBLE
CANNOT DUPLICATE	X	X	X	X	X				
RETEST OK	X	X	X	X	X	X	X		
TEST SYSTEM UNRELIABLE	X	X	X	X	X	X	X	X	
INCREASING TIME AND SKILL LEVELS	X	X	X	X	X	X			
NO CAPABILIT' WHEN TEST SYSTEM FAILS								X	X

Fiqure 4. Causes of Test and Maintenance Problems

Inadequate fault isolation—Each failure should be isolated to the specific failed replaceable unit. Deficient test and maintenance systems may isolate problems to the wrong unit or more commonly isolate a problem to a group of units. In the latter case, the maintenance technician has to replace each unit in sequence until the failure is corrected.

Incompatibility of test and test tolerances—Incompatibility is caused by independent (usually in a different timeframe) development of the testing of units at the intermediate and depot shop levels as opposed to the development of the onboard testing. The test equipment, the tests, the test conditions, and the test acceptance criteria can be and often are all different at the various test levels.

Out of specification but system still operates—This describes a condition in which a failure has caused a function to degrade outside its specific operating limits. The system is then technically failed because a portion of it no longer meets specification, but to the operator the system is still operating correctly. Another situation is one in which a component is out of tolerance but another component using the first one's output has margin to accept the condition and functions correctly.

Faults in BIT hardware—These faults take two forms. The first is when the BIT hardware fails so that it indicates a failure of some function when that function is not actually faulty. The operator sees the BIT indication and also the proper operation of the system. The second problem is when the BIT fails but continues to indicate a good condition even when a fault occurs. This condition will not normally be detected. When a failure of the tested function occurs, the operator may notice the loss of the function and no BIT indication.

Test data not accessible—In most onboard test systems the maintenance operators do not have access to the raw test data (BIT results) or are unable to interpret the data. When the automatic system fails to provide the correct isolation, there are no further resources available.

2.4 INTERRELATIONSHIP OF CAUSES

The causes of these problems are not all totally independent. For example, mode-dependent failures, environmental-dependent failures, intermittent failures, the condition in which the system is out of specification but still operates, and faults in the BIT hardware have been labeled false alarms. Mode-dependent failures and environmental-dependent failures may manifest themselves as intermittent faults.

There is considerable disagreement as to what constitutes a false alarm. Contractors tend to consider only design defects as false alarms; that is, errors in the circuit design or logic design that cause failure indications when certain operating conditions occur, but are independent of a specific unit and occur when operated within specified conditions. In theory there should be no design defects in a fielded system; but with the complexity of current avionics, problems continue to be worked out even after a system is fielded.

The users of systems consider false alarms to be failure indications for which no maintenance needs to be or can be accomplished or where maintenance that was accomplished was ineffective. There are other definitions of false alarms. Lear Siegler(1) considered that half the faults in BIT circuits constitute false alarms. This is based on the assumption that half the failures in the BIT circuits caused failure indications when there were none outside the BIT. Although this assumption may be appropriate for the pilot, it is inappropriate for maintenance, because although the other functions of the equipment are unaffected, the unit does contain a failure and does require corrective maintenance.

The views of the contractors and users are valid. By definition, a false alarm is an indication of a failure when none exists. By this definition alone, certainly the contractor's view is correct. These fault indications are built into the equipment as a result of oversights in the design. Maintenance cannot correct the problem. But what about the user's point of view? Certainly the false alarms defined by the contractors are included in those defined by the user, but there are other types of false alarms that are of concern. One example is when the equipment is operated in an extreme environment, beyond that specified for the equipment (for example, at high temperature), and begins to malfunction. The BIT correctly reports that the equipment is malfunctioning. During ground maintenance when the operating conditions are normal, the BIT no longer reports a failure. The maintenance technician is unable to duplicate the condition; and even under such duplicate conditions, no repair action is warranted. Any other unit is likely to operate similarly. In this case, the contractor says this is not a false alarm because the BIT correctly indicated something is wrong. The user, however, says this is a false alarm because no corrective maintenance is required.

Another example is when a failure is indicated in an extreme environment but not during normal operating conditions. This time the environment is not beyond that specified for the equipment, and the malfunction is caused by a degraded part whose operation is affected by the environmental condition. In this case, the symptoms look the same to the maintenance technician but there is a difference. The malfunction should not have occurred, and the problem can be corrected by replacing the bad part. This, then, is not a false alarm. The problem then is how to distinguish between this case and the previous case.

This discussion indicates the need to define two types of false alarms. The first are those failure indications for which no fault exists and the second are those failure indications for which no maintenance is required. The second category can be considered to include the first. The first category is wholly a function of the design and can be controlled directly, but the second category is a function of operation and operating conditions for which there is less control.

The examples given above for the environment-sensitive failure indications illustrate the close relationship between false alarms and intermittent faults. The first example is a false alarm, but the second is an intermittent fault. The problem in distinguishing between the two conditions is the lack of ability to measure the environmental excursions with respect to the requirement or to determine how several like units behave in a similar situation.

Another kind of failure that is difficult to categorize is what can be referred to as typical anomalous behavior of digital equipment. These are glitches that occur in operation due to noise or timing problems that are not totally eliminated from the design. Some contend that these are a natural consequence of digital electronics. Others claim that the problem is inadequate design and testing. BIT systems tend to be particularly susceptible to these types of glitches in that the BIT detects the problems and then latches the fault indication. The operational circuits, however, in most cases are tolerant of the glitches. Either there are error detecting and correcting circuits or the operation is a repetitive one where one bad result has no effect on the final process.

2.5 IMPACT OF PROBLEMS AND CAUSES

These various problems and their causes affect maintenance cost. The key elements of cost with respect to maintenance are labor and spares inventory. The problem of the CND's causes expenditure of additional labor trying to reproduce failure symptoms and may result in greater use of spares when unnecessary repairs are made in an attempt to correct the failure. For the RTOK's there is an unnecessary expenditure of both labor and spares involved in retesting good units. The problem of test systems being unreliable is more psychological than directly relatable to cost as has been stated before. The problem of increasing time and skill levels obviously affects labor costs. The last problem primarily affects labor costs in that more time is needed to work problems. There may also be an impact on spares for the last problem if inappropriate repairs are made.

The next problem to be addressed is the relative magnitude of the costs of the various problems. Various sources estimate that the rate of CND's and RTOK's constitute up to 90% of maintenance (see appendix E). An estimate based on hard analysis generally applicable to avionics has not been found. A reasonable estimate of the current rate of CND's and RTOK's is between 30% and 50%. Other problems contribute significantly less to the cost of maintenance. Correcting the causes of CND and RTOK problems also corrects the greatest numbers of problems caused by unreliable test systems and required time and skill levels.

The last problem—that of no capability when the test system fails—has two causes. The first, faults in BIT hardware, contributes little cost because faults in BIT hardware represent less than 10% of the failures and most of these are detected and isolated by the test system. The second cause is related to those failures for which the test system was not designed to detect or isolate. These constitute less than 5% of the problems in most systems. Because these problems contribute little to maintenance costs, correcting only the causes of CND's and RTOK's will return the greatest benefit.

The labor expended on CND problems is 4% to 22% of the organizational-level labor for typical systems.(2) These labor rates are significantly lower than the 30% to 50% given earlier. The labor expended on a CND occurrence is less than that expended to isolate and correct a failure.

To estimate the potential savings in labor realized by eliminating CND and RTOK problems, how labor distribution, and which elements are affected must be determined. Figure 5 shows distribution of maintenance time between organizational-level and depot-level maintenance and the distribution time between activities at these levels.(2) At the organizational level, elimination of CND's will reduce labor in all categories except remove and replace. Likewise, elimination of RTOK's at the depot level will reduce all labor categories except repair. Approximating the labor devoted to CND's at 15% and assuming that approximately the same percentage is required to pursue RTOK's at the depot level, the potential savings by eliminating CND's and RTOK's is:

Organizational level	15% of (45% - 6.75%) =	5.7%
Depot level	15% of (55% - 5.5%) =	7.4%
Total		13.1%

2.6 POSSIBLE SOLUTIONS

The following paragraphs offer ways to resolve the various causes of problems. For each problem, one or more approaches are introduced and discussed relative to difficulties or effects associated with the approach. These are offered as a lead-in to the development of the ITM requirements.

Mode of operation dependency—BIT and other tests could be designed so that they are independent of modes of operation. This appears to be difficult and would likely result in higher BIT development costs if feasible. Another approach is for the test system to associate failure indications with modes of operation. This requires the ability to duplicate operating conditions to verify the failure or rely on recorded information.

Environment dependency—The impact of this can be reduced by monitoring conditions that might affect operation, recording them, and then associating them to failure indications. The difficulties include defining what environmental conditions affect which

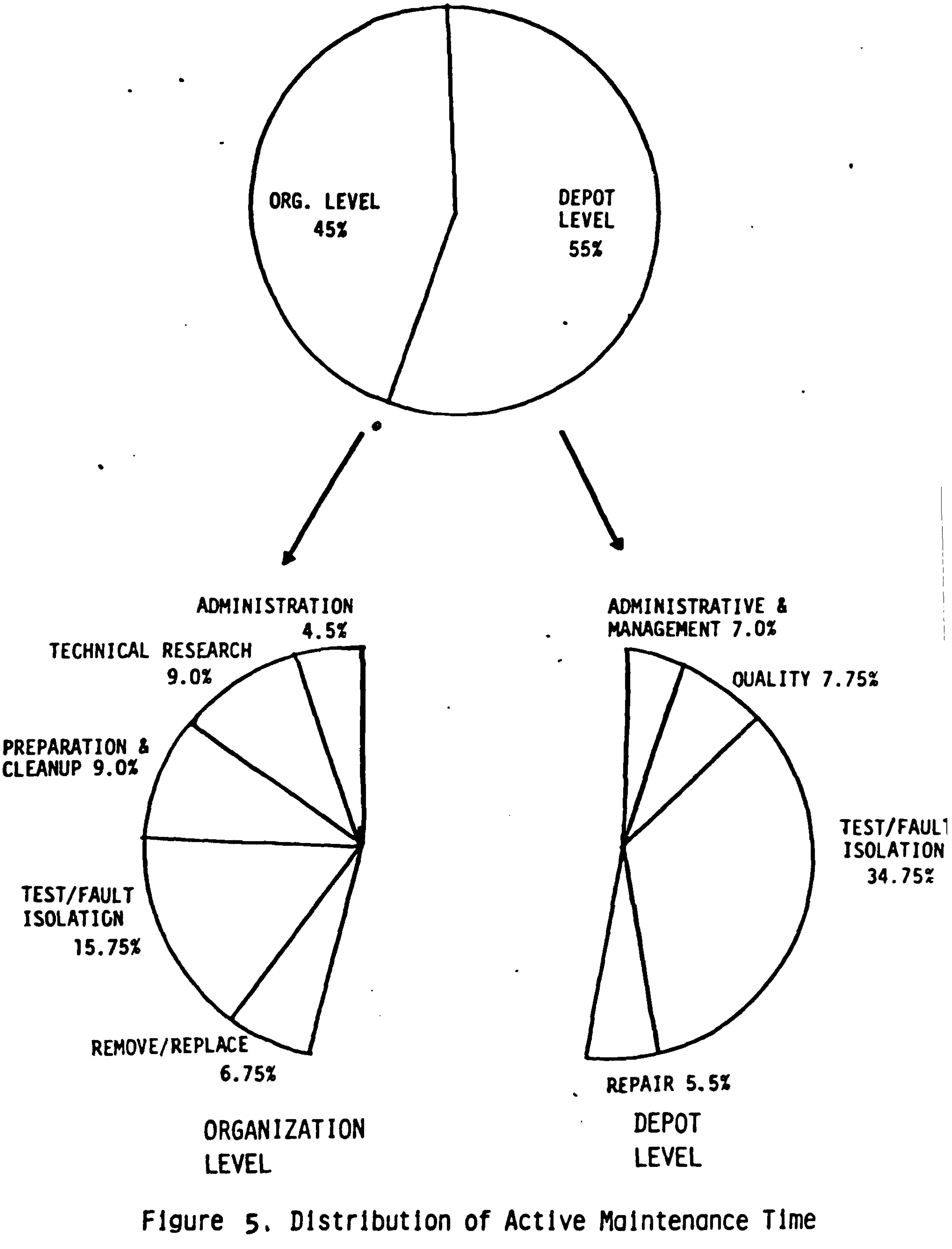


Figure 5. Distribution of Active Maintenance Time

hardware elements, how the conditions at the measurement point relate to the affected point, how to justify the measurement relative to specified levels, and how to duplicate conditions during retesting of the unit.

False alarms—To eliminate the false alarms due to design defects the only solution is a thorough design and testing effort. Because development cycles are already strained and failure modes are difficult to define and predict, there is significant difficulty implementing this solution. The subject has been discussed at length.(3)

Intermittent faults—The difficulty with resolving intermittent faults is determining when they affect operation and need to be corrected. Unnecessary failure indications must be isolated from the user and provide enough information to the maintenance technician to permit isolation of the failure. Current systems provide some filtering of intermittent faults (e.g., require two failure indications in succession or provide for a minimum time for the failure indication to be present). In general, these provide a general test for all or large classes of failure indications. What is needed is thresholds for number of occurrences, duration of occurrence, frequency of occurrences, or combinations of number, duration, and occurence for each test. Even with these, it is still a statistical problem to determine when unnecessary fault indications might occur and when faults might not get detected.

Inadequate fault isolation—In most cases, because the fault isolation diagnostic trees exhaust the combinations of test results, providing more isolation requires adding more test capability. Unfortunately the cost of adding test capability is an exponential function of failure coverage.(4) As shown in figure 6, 10% to 15% BIT is required to achieve a testability level (the product of fault detection rate and fault isolation rate) up to 95%. Beyond that, the fraction of BIT increases rapidly, approaching 30%, for a testability level of 98%. Systems being developed are on the knee of the curve. Any additional capability will be very expensive.

Incompatibility of tests and test tolerances—Solution of this problem requires change to program management at high levels. Most often the organizational-level test capability and the depot-level test capability are contracted for as two separate procurements, usually separated by a time lag and often procured from different contractors. This results in separate test equipment, test concepts, and test criteria. The

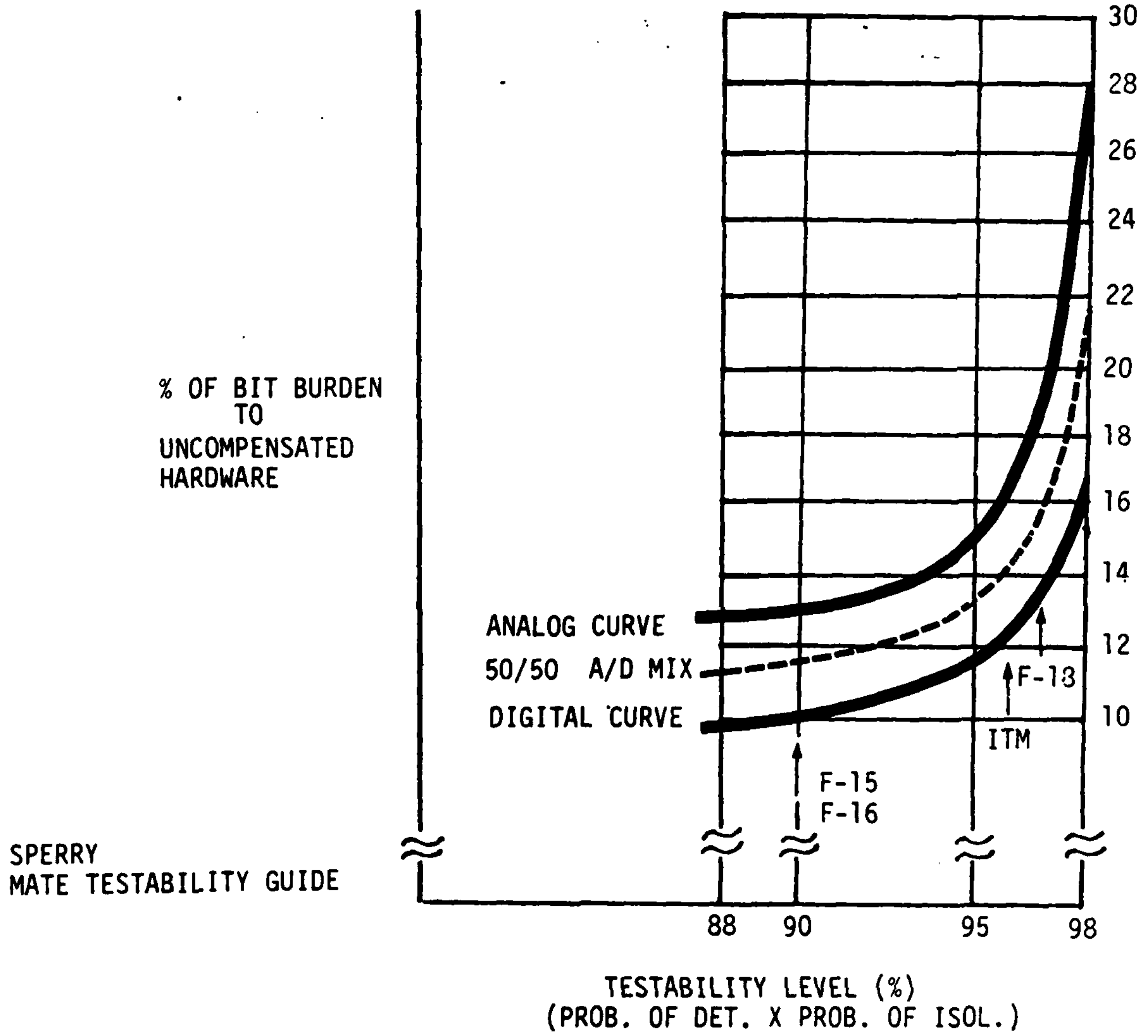


Figure 6. Organizational Level Testability To One LRU

obvious solution is to procure and develop all of the test capability for a system as a single effort. Another possibility is to develop standards for implementation of test capability that resolves the incompatibility that has been encountered. Effort has been directed toward this approach(4).

Out of specification, but system still operates—During this condition one element of a system is out of tolerence (i.e., failed), but another element tolerates the condition and the system continues to operate. This could be at the subsystem or LRU level, but the condition is more common at the circuit card or component level. Variability of electronic components and the influence of a good design practice of allowing some design margin contribute to this condition. These margins should be reduced, but it is probably inappropriate to legislate the elimination of design margins. For example, if part of the components need 1% regulation of power and some need 5%, a single 1% power supply will almost certainly be used instead of two supplies, thereby avoiding the cost and failure rate of an additional supply.

Faults in BIT hardware—Two approaches can be taken toward reducing the impact of faults in the BIT circuitry. One is to reduce the amount of BIT hardware and the other is to provide BIT for the BIT. In practice, combinations of both are being employed. BIT is becoming more software intensive with less dedicated BIT hardware, and the BIT hardware is being designed so that most BIT failures can be detected and isolated. It is not possible to detect all BIT failures (e.g., those that cause BIT to indicate good all the time); and if there were BIT for BIT, there could still be failures in the second level of BIT that would be undetected.

Test data not accessible—The solution to this deficiency is to make the data available to the maintenance technician. The difficulty involved is to decide which data are appropriate, how they are to be acquired and displayed to the technician, and how the technician can use the data. In general, designers have built into the automated diagnostics all conceivable ways that the available data can be used to diagnose all conceivable failures. Therefore, no new diagnostic information could be put into a technical order. It is possible, though, for a technician to be given the test data, the meaning of the individual data items, and what logic the system has already used in its analysis and then be able to accomplish additional diagnosis based on the symptoms of the failure.

2.7 FUTURE MAINTENANCE CONSIDERATIONS

ITM development must address anticipated as well as current maintenance considerations. Three anticipated considerations are the pilot's workload, packaging, and maintenance concept.

2.7.1 Pilot's Workload

There has been and will continue to be an ever increasing demand on the pilot as a system manager. This increasing demand must be considered in the development of ITM requirements. Elements of ITM design relevant to the pilot's workload are the controls needed to operate the system, the failure indications that occur, the system configuration and status information that is presented, and involvement required in the test and diagnostic process. All interfaces to the pilot must be minimized, especially in critical flight phases. The system must run automatically, and changes in configuration must occur without pilot interaction. Test sequence adjustments and diagnostics must be automatic. All failure information and system status information should be suppressed during critical flight phases except that affecting mission or flight capability or requiring corrective action.

With all this automation, however, there still should be provisions to permit the pilot to direct testing or make configuration changes if desired.

2.7.2 Packaging Trends

Electronics packaging will continue to evolve but the direction is certainly uncertain. Because maintenance capability is closely related to packaging design, it is important to consider how packaging will be done. With the increased use of microprocessor devices and VLSI circuits, more functions are being packaged in smaller assemblies. The three major directions in packaging are illustrated in figure 7.

When essentially the same functions are packaged in smaller boxes, the task of isolating to the LRU remains unchanged. If functions are integrated or combined in a smaller number of LRU's, as in Integrated Communication, Navigation, and Identification

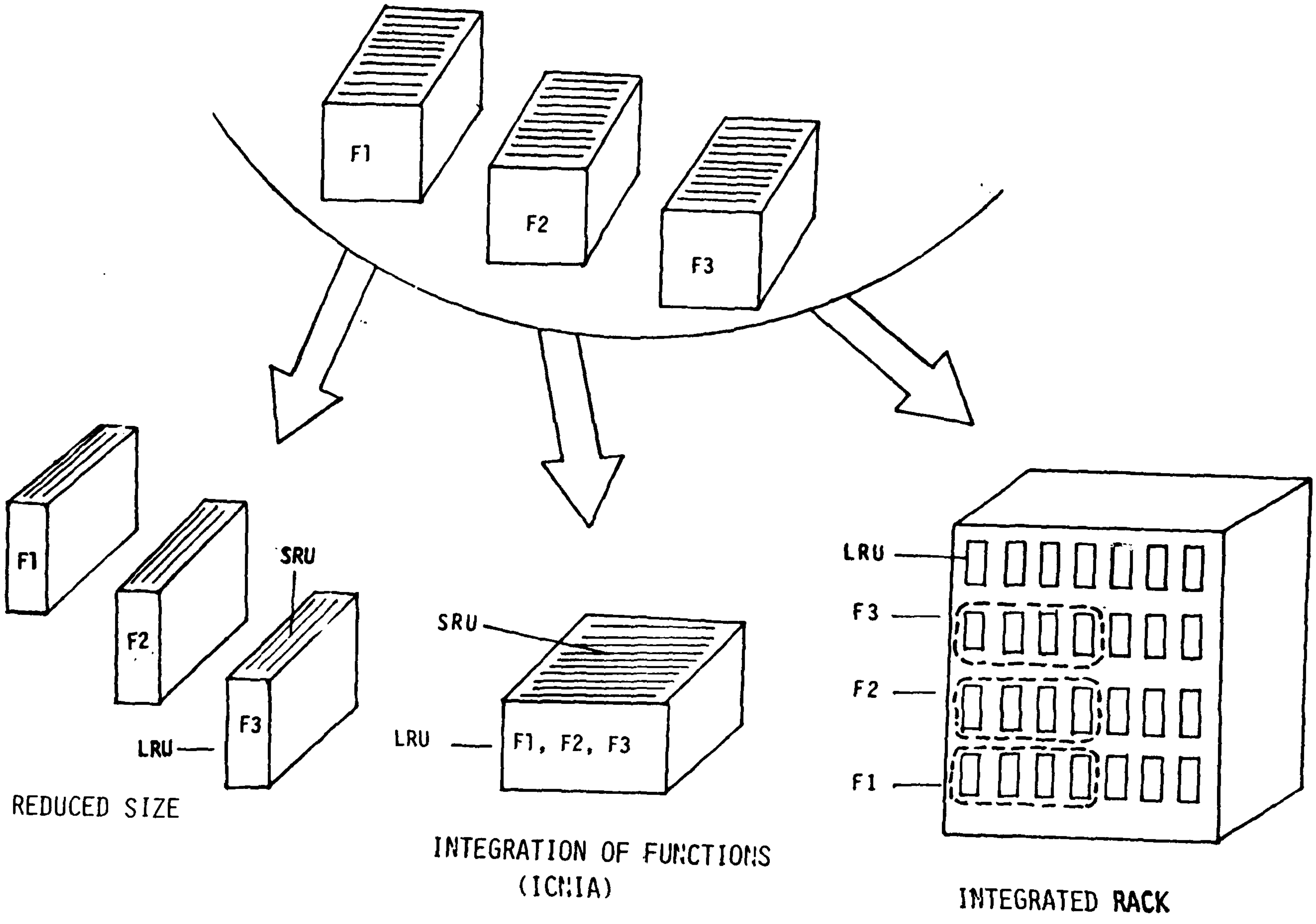


Figure 7. Future Directions in Packaging .

Avionics (ICNIA), the task of isolating to the failed LRU becomes easier in that more functions are combined in the same LRU.

A somewhat different packaging approach is to integrate numerous functions into a single rack in which printed circuit cards are packaged into replaceable modules. This presents a more difficult isolation problem since the LRU's constitute subfunctions. This integrated rack approach has problems that may delay incorporation into tactical fighters. The problems include interference between signals, cooling, exposure to environment during maintenance, and physical installation of the rack in a tactical fighter.

2.7.3 Maintenance Concept

ITM is to be developed for advanced tactical fighters. Tactical fighters operate from advanced bases with minimal support, in adverse weather conditions and with a short turn-around time between missions.

The traditional three levels of maintenance (organizational, intermediate, and depot) are summarized in table 1. Because one of the significant costs has been the cost of automatic test equipment (ATE) at the intermediate shop level, there has been a desire to implement two levels of maintenance as shown in table 2.

Discussions concerned with implementing a two-level maintenance concept suggest a benefit derived from replacement of circuit card, or shop replaceable unit (SRU), at the organizational level. The benefit is that lower cost units (as opposed to LRU's) will be in the repair pipeline. For the current packaging concepts this approach is undesirable for the following reasons:

a. Environment: Replacement of SRU's (circuit cards) requires opening of the LRU's, exposing both the SRU and the interior of the LRU to adverse environmental conditions.

b. Time: Due to extremely adverse packaging constraints there is already a burden on the repair time. Although systems are designed to reduce repair time, the nature of tactical fighters is such that LRU replacement is complicated by the LRU location, removal of access panels, and removal of the LRU. If there were an additional

TABLE 1. TRADITIONAL THREE-LEVEL MAINTENANCE

Level	Test level	Repair level	Test equipment	Location
Organizational	System	LRU (box)	System + AGE	Main operating base (or forward base)
Intermediate	LRU (box)	SRU (card)	ATE	Main operating base
Depot	SRU (card)	Piece part	ATE	Central depot

TABLE 2. CONCEPT OF TWO-LEVEL MAINTENANCE .

Level	Test level	Repair level	Test equipment	Location
Organizational	System	LRU	System	Main operating base (and forward base)
Depot	LRU SRU	SRU Piece part	ATE	Central depot

requirement for SRU replacement, either the LRU must be removed, the SRU replaced, and the LRU reinstalled; or the LRU must be designed to permit SRU replacement with the LRU installed in the system. The former procedure results in significantly increased maintenance time. The latter results in some increase in maintenance time and stricter packaging constraints.

The aspects of ITM development that enhance the capability to implement a two-level maintenance concept are (1) providing for complete on-board testing to the LRU level without the use of additional test equipment and (2) accurate isolation to the failed LRU. Of these, the latter is extremely important, because needlessly injecting serviceable parts into the repair pipeline has an even greater cost impact in the two-level maintenance scheme than it does in three-level implementation.

3.0 ANALYSIS OF CURRENT SYSTEMS

The development of the Integrated Test and Maintenance system (ITM) is based on the evolutionary improvement of built-in-test (BIT) techniques that have been incorporated into tactical and strategic aircraft over the past decade. To develop a system that takes advantage of the current state-of-the-art testing techniques as well as eliminating shortcomings of current systems, analysis was performed on test systems of aircraft most like the one to which ITM is targeted.

Based on mission role and aircraft similarities, the F-15, F-16, and F-18 were analyzed for various features of their onboard test and maintenance systems. The features compared were (1) concept, (2) pilot-crew interface, (3) intermittent handling, (4) data recording, (5) ground support equipment, (6) reconfiguration techniques, and (7) test concept.

In addition to comparing features, this section identifies the specified onboard test performance characteristics of each airplane and reviews the currently achieved testing results for intermittent failures, cannot duplicates (CND), and retest OK (RTOK). This includes onboard test results for the E-3A and EF-111A aircraft to provide a broader view of how onboard testing meets design objectives.

3.1 COMPARISON OF FEATURES

The following sections briefly describe the various features of the onboard test and maintenance systems and capabilities for the F-15, F-16, and F-18 tactical aircraft.

3.1.1 Concept

The onboard test and maintenance concepts for the F-15, F-16, and F-18 are described in the following paragraphs.

F-15—Each avionics set is responsible for its own BIT and must operate without depending on another set. The BIT of each system allows quick turnaround of the aircraft and bare-base operation by minimizing the ground support equipment.

Each independent subsystem is responsible for reporting subsystem status. This status is displayed by a set of indicators on the BIT control panel (BCP). There are subsystem-level test routines but no system-level test routines.

F-16—The F-16 uses a data-gathering computer referred to as the fire control computer (FCC) to provide an integrated capability to (1) collect and store failures from the avionic subsystems, (2) command avionic equipment to perform detailed individual operational checks and more thorough fault isolation within an avionic equipment set, and (3) display the resulting failures to the pilot or maintenance personnel through the pilot-crew interface.

Each subsystem detects and reports faulty and out-of-tolerance conditions to the FCC. This fault detection and isolation mechanization is contained solely within each individual subsystem. There are no system level test routines.

BIT is used to isolate failures in two ways, depending on the subsystem being tested. One is to provide a failed indication (test number), which in most cases can be used to isolate the failed LRU and the function that failed within LRU. The other is to provide a display that the operator must observe to verify proper test response. The test consists of a series of automatically sequenced display patterns that the operator must evaluate. From these patterns the extent of subsystem degradation is assessed and the faulty LRU identified.

Two fault-reporting schemes have been devised to ease pilot workload while providing adequate fault information for failure analysis. The maintenance fault list (MFL) contains detailed information for all reported faults and the pilot fault list (PFL) contains the same information for only those faults that are of interest to the pilot. In this way the pilot receives fault information that he may use to determine degradation in system performance.

F-18—The F-18 onboard test system is designed to provide the pilot with unambiguous displays of avionic and nonavionic system status without interfering with primary mission-essential functions. The information presented is derived from BIT mechanizations resident within each avionic equipment set and from nonavionic BIT (NABIT). In addition to the organizational-level fault detection and fault isolation displays, the status monitoring subsystem provides a recording capability used for fatigue strain, engine condition, and tactical information recording.

Through periodic and initiated BIT, failures are isolated and fault data are transmitted from each LRU to the mission computer. The mission computer gathers these avionic failures and engine and airframe failure data as well as consumable status data and distributes this information to—

a. The pilot to inform him of degraded operation and system status.

b. The maintenance crew for system analysis and repair.

c. A maintenance signal data recording set to preserve the information for later review.

d. A tactical reversion system to maintain the best available source of data for mission operation.

3.1.2 Pilot and Crew Interface

This section describes the controls and displays incorporated in each aircraft in support of onboard test and maintenance.

F-15—The BIT display group consists of the BCP and the avionics status panel (ASP) and indicates systems that have malfunctioned to the pilot and maintenance crew. The BCP provides a manual way to initiate interrupted BIT and indicates the results of the test. The ASP identifies, to the maintenance crew, the malfunctioning LRU and/or its location.

The BCP provides a manual way to initiate a detailed BIT and indicates test results. To accomplish this the BCP—

a. Houses the lights to indicate the system malfunction.

b. Contains switches to initiate interrupted BIT. Interrupted BIT is initiated by selecting the desired system and depressing the appropriate initiate button.

The ASP, located in the nose wheel well, identifies to the maintenance crew the malfunctioning LRU and/or its location. This panel—

a. Contains mechanical latching indicators to identify the location of the failed unit.

b. Requires that the indicators remain in the failed state even if the malfunction no longer exists.

c. Contains interlocking switches, which prevent the accidental initiation of interrupted BIT to those systems that endanger the crew, information, and/or equipment.

F-16—The controls and display group consists of a master caution light, specific caution panel lights, and the fire control navigation panel (FCNP). The master caution light and caution panel lights illuminate for all flight control failures and for catastrophic MUX-BUS avionics failures. The dual flight-control fail light and flight-control panel lights indicate failure conditions and failed function areas.

In the MUX-BUS system, the FCC and FCNP display MUX-BUS avionics failures through an alphanumeric readout. This digital display helps the pilot to determine the degree of degradation in each of the 12 avionic subsystems.

The FCNP is used to display the following information for each fault:

a. Subsystem of the failure that has been detected.

b. Degree of severity of malfunction.

c. Specific subsystem test number that failed.

d. Number of occurrences of that fault.

e. Time since fire control computer power-up of first occurrence of that fault.

F-18—The F-18 status monitoring equipment set is composed of a master monitor display (MMD), a maintenance monitor panel (MMP), a multifunction display, equipment fail and status indicators, and certain control interlocks.

The master monitor display is used to display status monitoring information. Cautions, advisories, and BIT messages are displayed to the operator delineating the failure or anomaly. The status of avionic subsystems is made available to the pilot by selection of the BIT display. Other features provided by the MMD include—

a. Interrogation of all subsystems simultaneously and display of the status of each.

b. Subsystem status including GO/NO-GO, in test, overheat, and degraded.

c. Inflight recognition of subsystem degration automatically displayed to the pilot.

The MMP is a digital readout device located for quick access in the nose wheel well. It stores and displays maintenance codes for failures or system anomalies detected by BIT during equipment operation. The nonvolatile MMP memory has a capacity to store 64 three-digit maintenance codes. All codes are stored under mission computer command with the exception of certain NABIT inputs, which under prescribed conditions may be commanded by the maintenance signal data recording set. The MMP has the following features:

a. Indicates servicing requirements to maintenance personnel.

b. Provides automatically:

   1. Numeric display of failed item by code.

   2. Mechanical latch that indicates that the unit has detected an internal MMP failure.

   3. Fluids low latch that indicates a low fluid level indication is stored in the MMP.

   4. Weapons system fail latch that indicates a weapon system LRU failure is stored in the MMP.

c. MAINTENANCE CODE DISPLAY button allows retrieval of sorted data from the MMP on the numeric display.

d. FLUIDS CHECK button allows fluid status check if low status is discovered. FLUIDS LOW latch is set.

e. BIT/RESET guarded button allows MMP-initiated BIT and clearing of stored data.

The multifunction display provides pilot cueing for mode failures of LRU's.

Equipment fail and status indicators are lighted displays that indicate specific failures within an avionic and/or nonavionic function.

There are certain control interlocks that must be satisfied to enable the initiation of self-test on the various avionic equipment. These interlocks are the switches on the ground power panel, the flight control set BIT consent switch, and the intertial navigation mode switch.

3.1.3 Intermittent Handling

This section describes how each aircraft addresses the latching and display of intermittent failures.

F-15—The avionic status panel's (ASP) mechanical indicators are latched at the detection of a failure. These indicators remain in the failed state even if the malfunction no longer exists. The location of the detected failure is identified directly below the latched switch. The ASP indicators can be reset only manually by using the RESET switch in the ASP. All indicators will be reset simultaneously.

The BCP displays all detected intermittent failures to the pilot or maintenance crew. Indicator lamps can be reset by the pilot or maintenance crew.

The BCP contains a RECALL lamp-switch, which reinstates previously indicated failures if the malfunction still exists. A newly detected failure will illuminate the faulty system light and recall any previously reset lights.

F-16—An intermittent failure is detected the same as all other F-16 failures. However, it appears one time with several occurrences rather than appearing numerous times on the maintenance or pilot fault list. The maximum number of occurrences recorded is nine. This limit is imposed by the FCNP display space. All intermittent failures are displayed on the FCNP.

F-18—To minimize intermittent failures resulting from the aircraft or operational environment or both, failure information is filtered by each piece of equipment so that the failure must exist for a fixed time before declaring a failure. Once the time limit has been exceeded, the failure information is transmitted to the MMD and MMP.

3.1.4 Data Recording

This section describes the methods used to record failure data and associated mission data, operating modes, or environmental data within the avionic and nonavionic systems.

F-15—There is no formal failure recording available on the F-15. However, latched failures are available for groundcrew review on the avionics status panel located in the nose wheel well. In addition, both the pilot and crew may view failures through the BCP.

F-16—Each subsystem detects and reports failures or out-of-tolerance conditions to the FCC. The FCC then collects, records, and reports the failure to either the pilot or maintenance crew.

The time of the first occurrence appears with each entry in the table. The number that appears refers to the minutes and tenths of minutes since the last turn-on of power to the FCC. This time, along with two special event entries, takeoff (TOF) time and landing (LND) time, separates the failures into preflight, inflight, and postflight categories.

Enough FCC memory is reserved to contain up to 17 maintenance fault list (MFL) entries. Thus, the MFL can contain TOF, LND, and up to 15 failure indications or up to 17 failures if they occur before the TOF and LND events. Should a greater number of failures be encountered during a flight, the list will consist of the first 17 entries, and subsequent entries will not be stored.

In addition, failure data are stored within single LRU subsystems when the subsystem contains its own memory. For these subsystems, the failed function data remain with the LRU and can be immediately interrogated at the intermediate shop and/or depot.

F-18—The F-18 records mission operational data in three system elements: (1) the mission computer, (2) the maintenance signal data recording set, and (3) the MMP.

The mission computer contains an inflight monitoring and recording module, which records aircraft strain data, failure information, and engine maintenance-related data.

The maintenance signal data converter samples input discrete and analog data for scaling, conversion to digital formal, and transmission to the maintenance data recorder and MMP. It provides up to 200 subsystem signals to the data recorder and MMP.

The maintenance data recorder is a hermetically sealed, plug-in four-track cartridge, which stores digital data on Kapton tape at the rate of 30,000 bits/sec. The capacity of the magazine is 718 blocks of 1,024 16-bit words, which gives it a capacity of approximately 12M bits.

The MMP is a digital readout device located for quick access in the nose wheel well. It stores and displays maintenance codes for the failures or system anomalies detected by BIT during equipment operation. Nonvolatile MMP memory has the capacity to store 64 three-digit maintenance codes.

3.1.5 Ground Support Equipment

This section describes the ground support equipment required to support onboard test and maintenance.

F-15—The one piece of test support equipment required by the F-15 is the flight line avionic test set. This is used to exercise the system and extract test data.

F-16—The F-16 requires a significant amount of ground support equipment to provide postflight analysis of failures detected.

A flight line test set used for the flight control system is used to aid in LRU failure isolation. This piece of equipment is necessary because the flight control system is nonstandard with MIL-STD-1553B and therefore cannot be operated on the MUX-BUS.

In addition, there is required a stores management set flight line tester used for voltage checks between the station and individual weapons.

Finally, a MUX-BUS tester is used to verify the integrity of communication paths within the MIL-STD-1553B links.

F-18—No ground support equipment required.

3.1.6 Reconfiguration Techniques

Reconfiguration is defined differently for different systems. Applications extend from redundant buses or equipment to types of tactical reversion and, finally, to elaborate fault tolerance systems. To be accurate, this review uses the terms and definitions used by the respective aircraft manufacturers.

F-15—There is limited automatic reversion for the avionics. An example of automatic reversion is in the failure of the inertial navigation system (INS). If the INS fails, the central computer automatically reverts to the attitude-direction mode using the attitude, heading, reference system (AHRS) for attitude and heading. However, for most systems the backup is a manual function.

F-16—There is a dual redundant data bus that allows communication on the alternate bus when one bus fails. If a failure occurs in the FCC, bus control is transferred to the INS. The only subsystem redundancy is in the stores management set.

F-18—Automatic tactical reversion is mechanized for the following tactical areas: flight aids, navigation, landing, and air-to-air and air-to-ground weapon delivery. The method of determining reversion is based in the equipment's ability to indicate invalid data through its BIT monitoring. When the equipment determines that a function has exceeded a predetermined threshold, the data derived from that function are immediately indicated as not valid. The mission computer, upon receiving this indication, reverts to the next best available source. This reversion is maintained as long as the data remain invalid from the primary source. The pilot is provided with appropriate display cueing only when a reversion results in some loss of capability or performance.

Three forms of degraded mode advisories are—

a. Reversion to an alternative data source of equivalent accuracy with no pilot cueing.

b. Reversion to an alternative data source of lesser accuracy with pilot cueing.

c. Removal of displayed data when no acceptable sources are available.

3.1.7 Test Concepts

This section describes the onboard testing concepts used in the F-15, F-16, and F-18. The three main categories of tests include continuous BIT (continually monitors the LRU signal for a value), interleaved BIT (intersperses test signals and replies among operational data), and initiated BIT (initiated by the pilot or crew and causes an interruption of normal operation). However, there are other testing methods described that either do not fit exactly into one of the above categories or are known by a different name. In those cases, we include a brief explanation of the test before describing its specific capabilities.

F-15—The F-15 uses the three types of tests mentioned in the preceding paragraph.

a. Continuous BIT: Continually monitors particular signals for value, logic, or presence at the LRU level.

Examples:
Voltage measurements of LRU power supplies.
Voltage measurements of logic levels.
Signal presence or absence on data bus.

b. Interleaved BIT: Automatically intersperses test signals and replies among operating signals so they do not interfere with normal equipment operation.

Examples: Timeshare compilation of testing algorithms.
Display tests between display sweeps.
Radar BIT pulses inserted between operational pulses.

c. Initiated BIT: Initiated by the pilot or crew and causes an interruption of normal operation of the designated system for the duration of the test. There are 19 initiated tests with 3 restricted to ground use only. Systems and tests that can only be interrupted on the ground are so indicated.

Initiated BIT is initiated by selecting the desired system and depressing the initiate button on the BCP.

Examples: Inserts artificial input and measures amplifier gain.
Drives displays to preset position.
Simulates servo error and measures servo correction.

Failure to pass any BIT test causes—

a. The appropriate indicator light to illuminate.

b. The appropriate equipment location indicators on the ASP in the nose wheel well to latch.

c. The avionics BIT light on the caution light panel to illuminate.

Interrupted BIT, except INS, is initiated by selecting the function on the BIT select knob and depressing the initiate button. All previously reset BIT lights and the avionics BIT light, if applicable, will illuminate when the initiate pushbutton in depressed. The associated light will blink during the test and extinguish at satisfactory completion of the test or illuminate steady for the test failure.

Most initiated tests require 2 sec or less to complete and some are in the 100 ms range.

F-16—The F-16 uses self-test and BIT to detect and isolate failures.

Self-tests are automatic noninterruptive performance tests that are either continuously or periodically performed with the results being monitored without disturbing normal system operation. Self-tests, however, do not wholly isolate the faulty subsystem LRU responsible for the detected failure conditions.

Self-tests follow two categories: (1) FCS tests and (2) MUX-BUS tests. FCS tests are used for testing such equipment as the FCC, accelerometers, and servos. MUX-BUS tests are used for MIL-STD-1553B-based systems such as the INS, stores management set, and the attack radar subsystem.

In the FCC, which is made up of analog subsystems, self-tests continuously perform channel comparisons by the use of 86 analog monitors. The MUX-BUS system is a digital system and provides two types of self-tests. There are continuous signal comparisons in addition to individual subsystem self-tests. Examples of MUX-BUS tests are:

a. Parity errors.

b. MUX-BUS A to B wraparound fail.

c. DMA timeout fail in INU.

d. Discrete alarms check in the FCC.

Self-tests support secondary failure reporting by providing to subsystems that contain their own memory test results to be used by maintenance personnel in postflight analysis.

BIT's interrupt normal equipment operation and/or require operator participation. An example of this is in the radar display where BIT makes use of several automatically sequenced display patterns that must be observed by the operator to determine whether a malfunction exists.

As with self-test, BIT provides two categories of system testing: (1) flight control system testing and (2) MUX-BUS system testing. Depending on the subsystem being tested, each is used to isolate system failures. One is to provide a failed indication (test number) that, in most cases, can be used to isolate not only the failed LRU but also the function that failed within the LRU. The other is to provide a display that the operator must observe to verify proper test response. In addition, the MUX-BUS system allows for automatic isolation of the malfunction to an LRU.

There is one BIT for each major subsystem with a total of eight tests available. These BIT's are used to further isolate failures detected by self-test.

F-18—Avionics BIT is implemented within each LRU avionic set to provide fault detection and fault isolation. In most instances, two types of BIT are provided, periodic and initiated. Periodic and initiated BIT are discussed in a later paragraph.

Two forms of BIT derived data are supplied to the mission computer to generate the display of system anomalies. One form is validity information associated with selected data. This identifies whether the data are valid or invalid. The validity information is generated in real time and if a sampled function fails or exceeds its predetermined threshold, it will immediately be indicated as invalid. The mission computer uses validity information to automatically reconfigure the weapon system to provide tactical reversion.

The second form is the equipment failure information, which identifies failed LRU's.

Periodic BIT automatically begins upon equipment power application. It provides a failure detection capability less than that provided by initiated BIT since it must not interface with normal equipment operation. Periodic BIT provides complete current status of all avionic equipment that interfaces with the mission computer to the pilot. Currently there are eight possible status messages capable of being displayed to the pilot. The mission computer uses these eight messages and assigns a maintenance code to the MMP for the maintenance crew. There are currently over 150 possible maintenance codes.

Initiated BIT is a more rigorous version of periodic BIT, which interrupts the normal operation of the equipment under test. As with periodic BIT, initiated BIT performs validity information and equipment failure tests.

Initiated BIT displays status messages identical to those used for periodic BIT and are displayed as each equipment set enters, performs, and completes its BIT routine. The results of these tests are displayed in the pilot's cockpit display and on the maintenance monitor panel for the maintenance crew.

The length of time required for initiated BIT varies from 1 sec to 38 min with the majority of the possible 24 initiated BIT routines requiring from 5 to 25 sec. With the length of time required to perform certain tests being of a long duration, those tests are restricted to ground use only.

Maintenance BIT is provided to allow selection of unique operator participation tests, special data displays, and special equipment calibration routines. These onboard test routines are available only on the ground by selection of a MAINT button on the MMD. Maintenance BIT provides special options to be made available for seven systems. Maintenance BIT—

a. Provides more detailed information on a detected failure or initiated BIT to be displayed.

b. Provides special testing of several aircraft switches.

c. Provides the capability to perform a complete flight control maintenance BIT with operator participation tests.

d. Provides memory inspect, which allows the maintenance operator to inspect the memory contents of selected computers by use of upfront controls and cockpit displays.

3.2 SPECIFIED ONBOARD TEST PERFORMANCE

The specified onboard test performance characteristics of the F-15, F-16, and F-18 have shown an evolutionary improvement since the design and development of the first onboard test systems. Each aircraft has addressed onboard testing by assigning diagnostic specifications that are generally in the range of 90% to 95% probability of automatic (or semiautomatic) fault detection and isolation. False alarm rates have generally been specified to fall within the range of 1% to 2% percent of failures detected.

For the F-15, each newly developed avionics system contractor furnished equipment (CFE) had to detect 95% of all possible failures. Each system was also required to isolate 95% of detected failures to the LRU. The false alarm rate requirement could not be determined.

The F-16 avionic subsystems must have the capability to detect and report malfunctions and out-of-tolerance conditions that indicate that equipment performance is below an acceptable level by employing self-test and BIT. Self-test, which is to operate in all modes except OFF, is specified to be capable of detecting 95% of all malfunctions and out-of-tolerance conditions, with a design goal of 99%. For false alarms, not more than 1% of all indicated failures are allowable as false alarms. The combination of self-test and BIT must be capable of isolating to the failed LRU a minimum of 95% of the detected malfunctions and out-of-tolerance conditions.

BIT is implemented within each F-18 CFE avionic set to provide fault detection and fault isolation. Two types of BIT are employed, periodic and initiated. Periodic BIT provides a failure detection capability that is somewhat less than that provided in initiated BIT because it must not interfere with normal equipment operation. The BIT requirement for each equipment set is slightly different, but overall the failure detection capability is 98% in operator initiated BIT and 90% in periodic BIT. Fault isolation is 99% of the detected failures.

A combined summary of F-15, F-16, and F-18 fault detection, fault isolation, and false alarm rates is illustrated below.

	F-15	F-16	F-18
Fault detection	95%	95%	98%[a] 90%[b]
Fault isolation	95%	95%	99%
False alarms	---	1%	1%

[a] Initiated BIT.

[b] Periodic BIT.

3.3 CURRENTLY ACHIEVED TESTING RESULTS

The previous discussion centered on the specified diagnostic probabilities of fault detection, fault isolation, and false alarms. However, after much review, it was observed that the actual performance of onboard testing was much less than that specified at the time of design. This section will concentrate on onboard testing results obtained from reports, briefings, and workshops for the F-15, F-16, E-3A, and EF-111A(3). The F-18 is too new for sufficient testing performance data to be available.

F-15—No specific fault detection or fault isolation figures could be obtained to verify the diagnostic capability of BIT. However, studies of F-15 test and maintenance techniques are available to give a clear view of the performance of diagnostic testing. The conclusion was that at the time the study was finished, "F-15 test systems (BIT) is not very reliable and is often ambiguous."(25) Some of the reasons given for that conclusion are based on the following examples:

a. "Given a fully operational aircraft that had been found to be without equipment failures, a cable was removed at random from the subsystem and the aircraft test system could not isolate to the 'failed' LRU."

b. "A second failure was simulated and again the BIT system could not isolate to the failure."

c. "With the test system not able to isolate a failure to a single box, the maintenance technician had to remove three boxes to find the failed LRU."

F-16—Thirteen F-16 aircraft were monitored over an 18-month duration, which included 2,899 sorties and 3,825 flying hours. Both the FCS and MUX-BUS were monitored and categorized separately. A few comments are necessary to understand the particulars of each system.

a. The flight controls data (part of a fly-by-wire system, which is totally electronic) are not affected by data from interactions among other systems.

b. Fault detections in the flight controls system are presented to the pilot who must record the fault indications, as opposed to the MUX-BUS, which records the failures in memory for later readout. As a result, some of the failures (particularly intermittents) are missed in the analog flight control system.

c. The MUX-BUS incorporates twelve subsystems.

Table 3 shows the results of the self-test and BIT performance for the F-16 flight control system (a quad-redundant system). These results are as seen by the contractor and the Government. It was agreed that both reliability and maintainability are improved the more the aircraft is flown.

Similar results are shown in table 4 for the F-16 MUX-BUS. The fault reporting on the MUX-BUS does not include failures of input devices to the units on the MUX-BUS (e.g., angle of attack unit as an input to the air data computer (ADC).

A review of the figures indicates a large disparity in diagnostic performance as seen by the contractor and the Government. To understand this conflict in interpretation, we must look at the raw data that were used in this evaluation. This is illustrated in figure 10.

The terms associated with the evaluation process are defined below.

Special Category

Engineering deficiencies—Self-test and BIT reported failures of system problems induced by design deficiencies (hardware and/or software) upon which organizational maintenance actions had no effect.

No trial—A report of a failure that is not worked (not followed by maintenance action) for a variety of reasons such as judgment, lack of confidence in self-test and BIT, operator induced, or lack of technical training.

CND

A reported failure that is worked by maintenance but cannot be verified by self-test, BIT, or other methods.

Addressable Failure

Failures that self-test and BIT was designed to detect and isolate that were corrected by maintenance.

TABLE 3. F-16 FLIGHT CONTROL SELF-TEST AND BIT PERFORMANCE

	Results		Rating	
Measure of effectiveness	As contractor sees it	As user sees it	As contractor sees it	As user sees it
Fault detection	100%	83%	Excellent	Deficient
Fault isolation	92%	73.6%	Excellent	Deficient
Cannot duplicate	-----	17%	-----	Deficient
Retest OK	-----	20%	-----	Deficient

TABLE 4. F-16 MULTIPLEX BUS RESULTS

	Results		Rating	
Measure of effectiveness	As contractor sees it	As user sees it	As contractor sees it	As user sees it
Fault detection	90%	49%	Satisfactory	Deficient
Fault isolation	93%	69%	Satisfactory	Deficient
Cannot duplicate	-----	45.6%	-----	Deficient
Retest OK	-----	25.8%	-----	Deficient

After reviewing the raw data, it was noticed that in the flight control system only, a few no-trials and engineering deficiencies exist. This is because the flight control system lacks a totally automatic nonvolatile reporting svstem as in the MUX-BUS. Therefore, some data were lost during debriefing because aircrews failed to write up failures that occurred before and/or during flight. These failures did not affect flight performance and disappeared during subsequent retests.

Figure 8 shows that the contractor claims 100% of all failures were detected, and the Government claims a failure detection rate of only 83%. This lack of agreement is based on the failure to interpret the results in the same manner. As an example, the total number of failures in the flight control system is 200 (166 addressable failures + 34 CND's). The Government insists that the inability to duplicate a fault decreases the fault detection percentage (166 addressable failures : 200 total failures = 83%). The contractor counters that because knowledge of the failure exists, the failure was detected (166 addressable failures + 34 CND's = 200 failures. 200 failures : 200 total failures = 100%). The same arguments are waged over the fault isolation percentages as well as the evaluation of onboard testing for the E-3A and the EF-111A.

E-3A—Nineteen E-3A aircraft were monitored over an 18-month period (July 1978 to December 1979), which included 791 sorties and 6,205 flying hours. Only the surveillance radar of the E-3A is included in this analysis. Fault isolation in the E-3A is offline, requiring transfer of the diagnostic programs for execution. Program results are shown in table 5, indicating the effect of CND's on detection and nondetection percentages as viewed by the Government and the contractor. Also shown are the effects of RTOK's on isolation precentages as viewed by the two parties. As can be seen, CND and RTOK requirements were not imposed on the contractor for the E-3A program. A few observations of the results not apparent from the figure below are (1) the actual autoisolation percentage is shown to be between 34% and 49% and (2) the RTOK rate was essentially the same whether fault isolation was automatic or manual (about 30%).

EF-111A—Review of the EF-111A aircraft onboard test capabilities (table 6) was modest compared to the F-16 and the E-3A effort (one aircraft, 5 months, 36 sorties, and 261 hours). However, this phase allows us to observe preliminary results before a more thorough undertaking whose results will be available later.

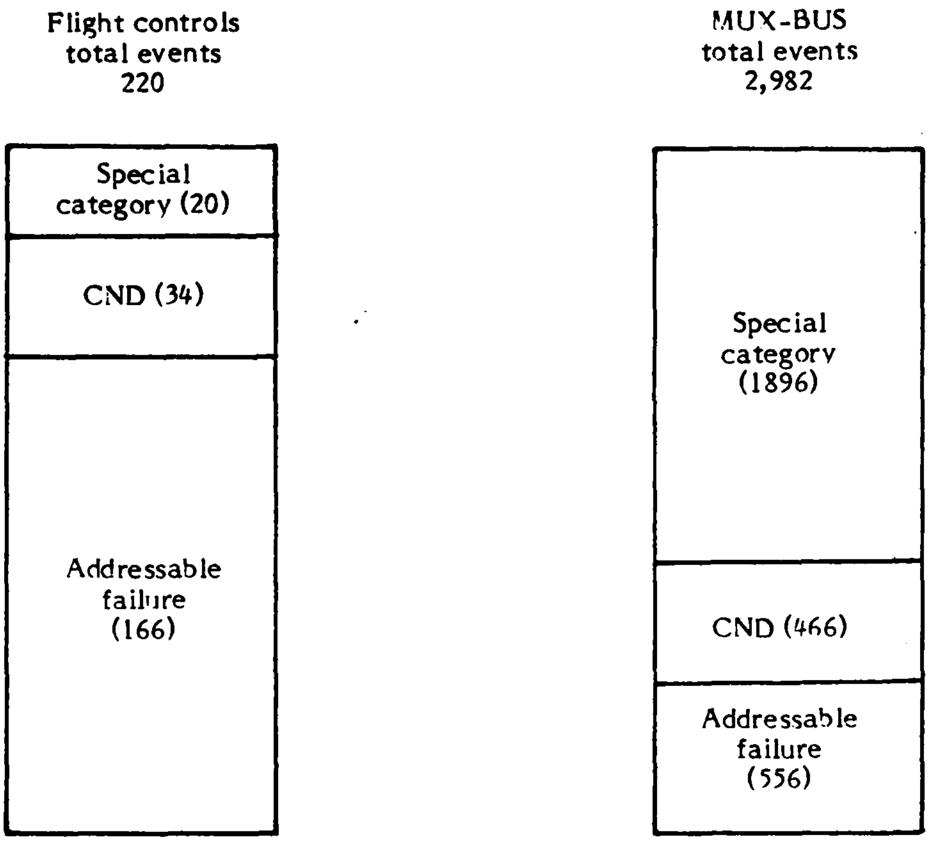


Figure 8. F-16 Raw Data Base

TABLE 5. E-3A RADAR BIT/FIT PERFORMANCE

	Results		Rating	
Measure of effectiveness	As contractor sees it	As user sees it	As contractor sees it	As user sees it
Fault detection	98%	74%	Excellent	Deficient
Fault isolation	49%	34%	Deficient	Deficient
Cannot duplicate	-----	17%	-----	Deficient
Retest OK	-----	20%	-----	Deficient

TABLE 6. EF-111A BIT/BITE PERFORMANCE

	Results		Rating	
Measure of effectiveness	As contractor sees it	As user sees it	As contractor sees it	As user sees it
Fault detection	100%	62%	-----	-----
Fault isolation	88%	71%	-----	-----
Cannot duplicate	-----	38%	-----	-----
Retest OK	-----	19.2%	-----	-----

F-18—Onboard test performance results are not yet available for analysis due to the verdancy of the program. Program results are expected about the first quarter of 1983 and will be presented in the final report.

4.0 DEVELOPMENT OF ITM CAPABILITIES

4.1 COMPILATION OF CANDIDATE ITM CAPABILITIES

A list of ITM capabilities was compiled by—

a. Reviewing the ITM capabilities description provided in appendix D of the statement of work.

b. Reviewing SSA00501001, mission demonstration specification for the advanced system avionics, operational readiness mission.

c. Reviewing related avionics testing reference material.

d. Reviewing recent tactical fighter onboard test capability.

e. Drawing upon Boeing's experience with onboard test systems.

The compiled list is shown in table 7 with an indication of the flight phases to which each capability is applicable. The following provides a brief description of each capability.

Establish operational readiness—Applicable to the preflight phase, this capability combines BIT and tests of operational information to determine if the system is capable of performing the required mission functions.

Continual fault monitoring—Applicable to the inflight phase, this capability combines BIT, reasonableness tests of operational data, and special test techniques to continually monitor for faults. Some tests may be continuous, some may be periodic, and some may be aperiodic.

Fault detection (X%)—This is a specification of the percentage of faults that must be detected. Detection of faults is accomplished by a combination of BIT, reasonable tests of operational data, special test techniques, and operator observance of displays. The percentage requirement may change as a function of flight phase because different test resources and conditions are available in each flight phase.

Support reconfiguration—ITM interfaces with the fault tolerant functions of Operational Flight Program (OFP) in the following way. When a fault is detected, it is isolated to a level where it can be determined if the fault can be circumvented by a

TABLE 7. ITM CAPABILITIES

	Preflight	Inflight	Postflight
Establish operational readiness	X		
Continual fault monitoring		X	
Fault detection (X%)	X	X	X
Support reconfiguration	X	X	
Minimal pilot interface		X	
Fault isolation to support reconfiguration	X	X	
Fault isolation to LRU (X%)	X	X	X
Fault isolation to SRU (X%)			X
Maintain system status	X	X	
Fault data recording	X	X	X
Fault record processing			X
Report generation			X
Manual test interaction	X		X
Manual isolation			X
Manual BIT Interrogation			X
Automated technical orders	X		X
No additional equipment	X	X	X
Aircraft power only	X	X	X
Intermittent handling	X	X	X
False alarm requirement	X	X	X
Environmental data recording		X	
Multiple fault isolation			X

reconfiguration action. The fault tolerance function actually defines the reconfiguration requirements and implementation. After the reconfiguration occurs, the fault tolerance function transfers to ITM the new configuration status, and ITM retests the system. ITM verifies that the reconfiguration was accomplished and that the system is capable of performing the required mission functions.

Minimal pilot interface—The pilot interfaces to ITM via the mission keyboards and displays. During the inflight phase of the mission the interaction between ITM and the pilot should be reduced to a minimum. The pilot should be interrupted only when a fault impacts mission success.

Fault isolation to support reconfiguration—To support reconfiguration, faults need to be isolated to a level that the fault tolerance function can determine whether a fault can be circumvented by reconfiguration and how the reconfiguration is to be accomplished. In many cases it is not necessary to isolate to the specific LRU that is faulty. For example, when one of the buses is faulty, the immediate need is to switch to the redundant bus. It is not necessary during the mission to isolate to the specific bus component.

Fault isolation to line replaceable unit (LRU) (X%)—The LRU is a hardware element that is replaced during flight line maintenance actions. These typically are avionics-rack-mounted modules, control panels, and wiring and rack components. The capability provides the identity of the faulty component without the need for the technician to do any isolation.

Fault isolation to ship replaceable unit (SRU) (X%)—The SRU is a hardware element that is replaced in an intermediate or depot shop. Its physical nature is such that it is not suitable for replacement in the flight line environment. Typical SRU's are circuit cards or submodules of an avionics module. The capability provides identity of the faulty SRU to reduce intermediate or depot repair effort.

Maintain system status—This capability provides for continuous, online status by function and by individual equipment items. The information is maintained by the Integrated Testing and Maintenance system (ITM) for use by the operator (pilot or technician) and the automated test functions for control of testing.

Fault data recording—Fault data are recorded on the mission recording media for postflight processing to assist in maintenance and for offline processing for additional diagnosis and maintenance history. The types of data that are recorded are the fault data themselves, related prior events, reconfiguration data, system status, intermittent fault data, and isolation results.

Fault record processing—In the postflight phase the fault data recorded in flight are processed to provide fault diagnosis, identify units to be replaced, and provide more extensive fault isolation. Another aspect of fault record processing occurs offline where additional diagnosis can be accomplished and maintenance history can be maintained.

Report generation—This capability could vary from providing maintenance information in an order and format for direct transfer to maintenance report forms to the actual printing or formating for printing the actual report forms.

Manual test interaction—This provision involves the operator (pilot or technician) in the test process. In the preflight phase this would include activities like checking the displays via test patterns and checking the control switches. In the postflight phase manual interaction may involve switching equipment through operational modes to assist in isolation.

Manual isolation—In order to provide 100% diagnostic capability, ITM must include the capability to isolate those faults that exceed the automated isolation. This includes the process of isolation and any provisions incorporated to assist in isolation.

Manual BIT interrogation—This is one of the provisions to assist in manual isolation. It provides the technician the capability to access raw test data through the ITM and crew interface.

Automated technical orders—In the preflight mode this is primarily computerized checklists displayed to the pilot. In the postflight mode this includes not only computerize diagnostic procedures displayed to the technician but procedures that could be extended to include maintenance messages providing remove, replace, or adjustment procedures.

Aircraft power only—Using aircraft power eliminates the need for any ground power or portable power facilities and reduces maintenance test time.

Intermittent handling—Intermittent faults must be detected, isolated, and corrected; but their occurrence must not be allowed to interrupt the mission. An intermittent fault that does not adversely affect mission data should not cause a reconfiguration of system resources. Intermittent faults can be filtered in fight using frequency and duration thresholds and can be isolated during postflight by comparing intermittent fault occurrences with operational data and flight conditions.

False alarm requirement—This specifies a maximum acceptable false alarm rate. False alarms dilute system resources in that, in flight, unnecessary reconfigurations may occur; and on the ground, unnecessary maintenance may occur.

Environmental data recording—This provides for inflight recording, in addition to fault data, any environmental information that may be useful in correlating to the fault data.

Multiple fault isolation—If multiple faults occur, this capability allows for isolation of each fault by recognizing and reacting to possible interrelationship of fault symptoms.

4.2 DETERMINATION OF ESSENTIAL CAPABILITIES

The next step in the task was to examine each of the capabilities and determine which are essential to a cost effective implementation of ITM and which were desirable but not essential. The results are provided in table 8. Included is a ranking of the desirable capabilities to indicate the order in which they might be considered for incorporation.

The following is a justification for why each of the desirable capabilities was not considered essential. Also provided is an explanation for the ranking.

Fault isolation to LRU—Isolation to the LRU in the preflight and inflight phases is not essential because no additional utility is gained above the ability to isolate to support reconfiguration. When considered in relation to the other desirable capabilities, however, it is the most important. Because one of the major goals of ITM is support of flight-line

TABLE 8. ESSENTIAL OR DESIRABLE ITEM CAPABILITIES

	Preflight	Inflight	Postflight
Establish operational readiness	E		
Continual fault monitoring		E	
Fault detection (X%)	E	E	E
Support reconfiguration	E	E	
Minimal pilot interface		E	
Fault isolation to support reconfiguration	E	E	
Fault isolation to LRU (X%)	D(2)	D(1)	E
Fault isolation to SRU (X%)			D(3)
Maintain system status	E	E	
Fault data recording	E	E	E
Fault record processing			E
Report generation			D(7)
Manual test interaction	E		E
Manual isolation			E
Manual bit interrogation			E
Automated tech orders	D(5)		D(6)
No additional equipment	E	E	D(4)
Aircraft power only	E	E	E
Intermittent handling	E	E	E
False alarm requirement	E	E	E
Environmental data recording		D(8)	
Multiple fault isolation			D(9)

E = Essential
D(n) = Desirable (rank)

maintenance, any advance isolation data are very valuable. In flight, the data can be used on the return leg to provide advance preparation for repair actions. In the preflight environment the capability is less important because either faults will not affect the mission and isolation can occur later or the mission will be aborted and the postflight conditions will take effect where LRU isolation is essential.

Fault isolation to SRU—By definition an SRU is not replaced on the flight line. Therefore, it is not an essential ITM capability to isolate to the SRU. But again, since cost effective maintenance depends on good isolation, it is desirable to advance isolation provided for shop repairs. This is less desirable than isolation to LRU, which directly supports flight-line maintenance.

No additional equipment—Not having additional equipment may be unavoidable in the postflight phase. Equipment may be needed to support required isolation capability or data processing that would adversely effect the system if it were built-in. In desirability this ranks above the next two capabilities, which are substitutes for documentation.

Automated technical orders—This capability is not essential because it is a trade between documentation and computer resources. Documented technical orders provide additional capability because they have pictorial information, but they are more cumbersome to use than automated technical orders. Because the postflight technical orders include repair and adjustment procedures, the computer resources required are significant. For this reason, it is considered a lower priority than for preflight automated technical orders.

Report generation—Report generation is a substitution for computing resources for documentation. This is a lower priority because the efficiency gained is less than in automating the technical orders.

Environmental data recording—The types of data that could be monitored and recorded are—

a. Equipment internal temperatures.
b. Equipment ambient temperature.
c. Cooling airflow.
d. Vibration.

e. Acceleration.

f. Electromagnetic interference (EMI).

g. Humidity in equipment bay.

h. Pressure in equipment bay.

These are arranged in approximately the order of usefulness. The value of environmental data recording is in diagnosis of certain environmentally related faults. The data can be correlated to fault data in postflight and offline processing for additional diagnostic capability. Considering that only a portion of intermittent faults are involved and the cost in terms of added sensors, data collection hardware, onboard processing and offline processing is great, an extensive environmental data collection system is not warranted. There is, however, justification for a limited number of temperature and cooling air measurements.

Multiple fault isolation—The capabilitv to isolate multiple faults is the least desirable capability for the following reasons:

a. The occurrence of multiple faults should be low due to high equipment reliability, relatively short mission duration, and the intent to maintain the system in a full-up condition.

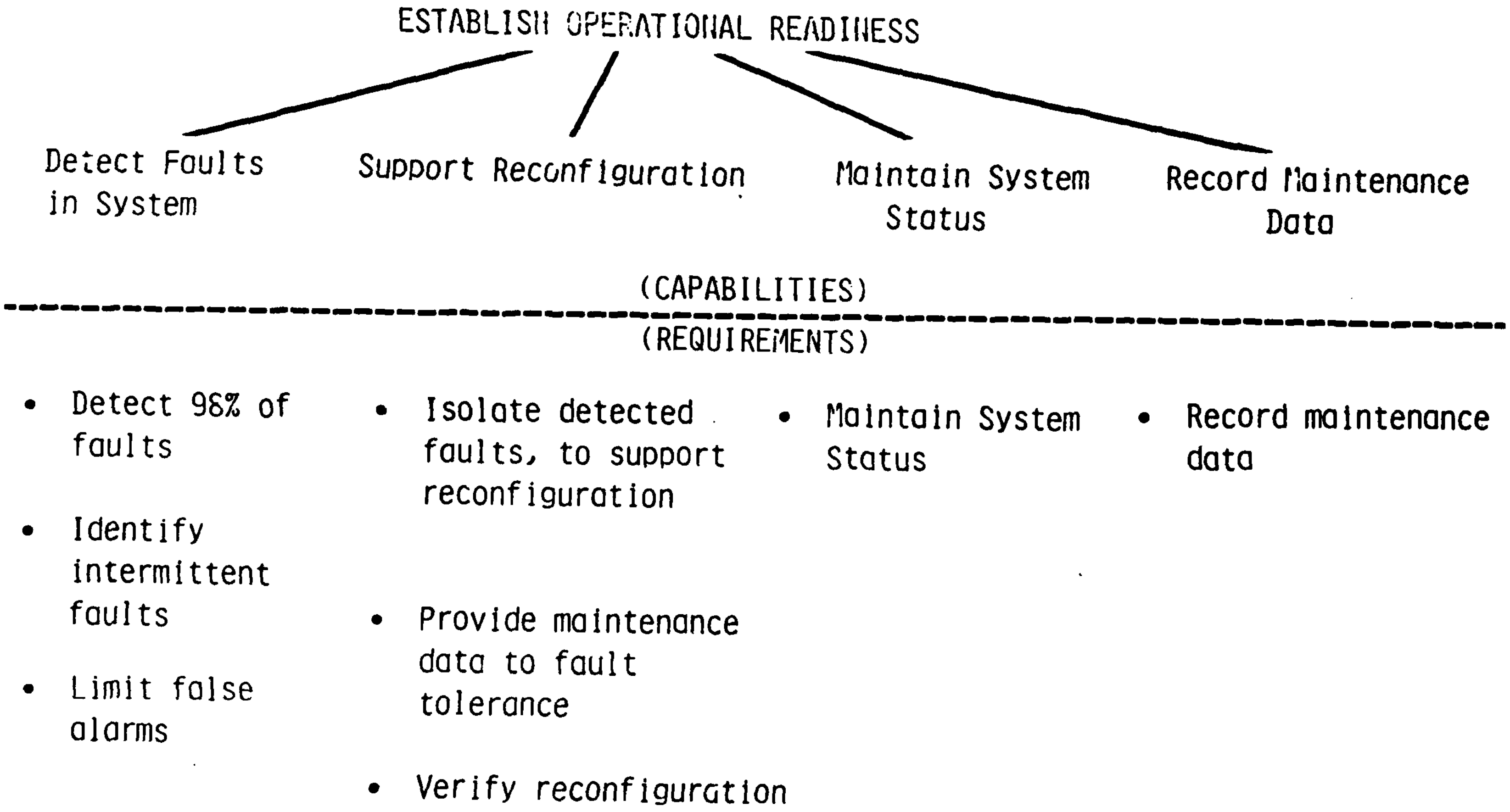
b. Even when multiple faults occur, the majority will be in disjoint subsystems or functions and can be isolated as single faults.

c. Extensive resources are required to isolate multiple related faults.

While it is recognized that, in practice, faults are occasionally allowed to accumulate, this still does not justify a multiple fault isolation capability as an essential capability.

4.3 SUMMARY OF ITM CAPABILITIES

To direct development of ITM system requirements from the top down, the list of ITM capabilities needed to be organized to establish the interrelationships of the various capabilities. In doing this, it was determined that some are more appropriately considered as system requirements and some, ground rules. The results are summarized in figures 9 through 11. The foremost capabilities of ITM in the three flight phases are—



ESTABLISH OPERATIONAL READINESS

Detect Faults in System

Support Reconfiguration

Maintain System Status

Record Maintenance Data

(CAPABILITIES)

(REQUIREMENTS)

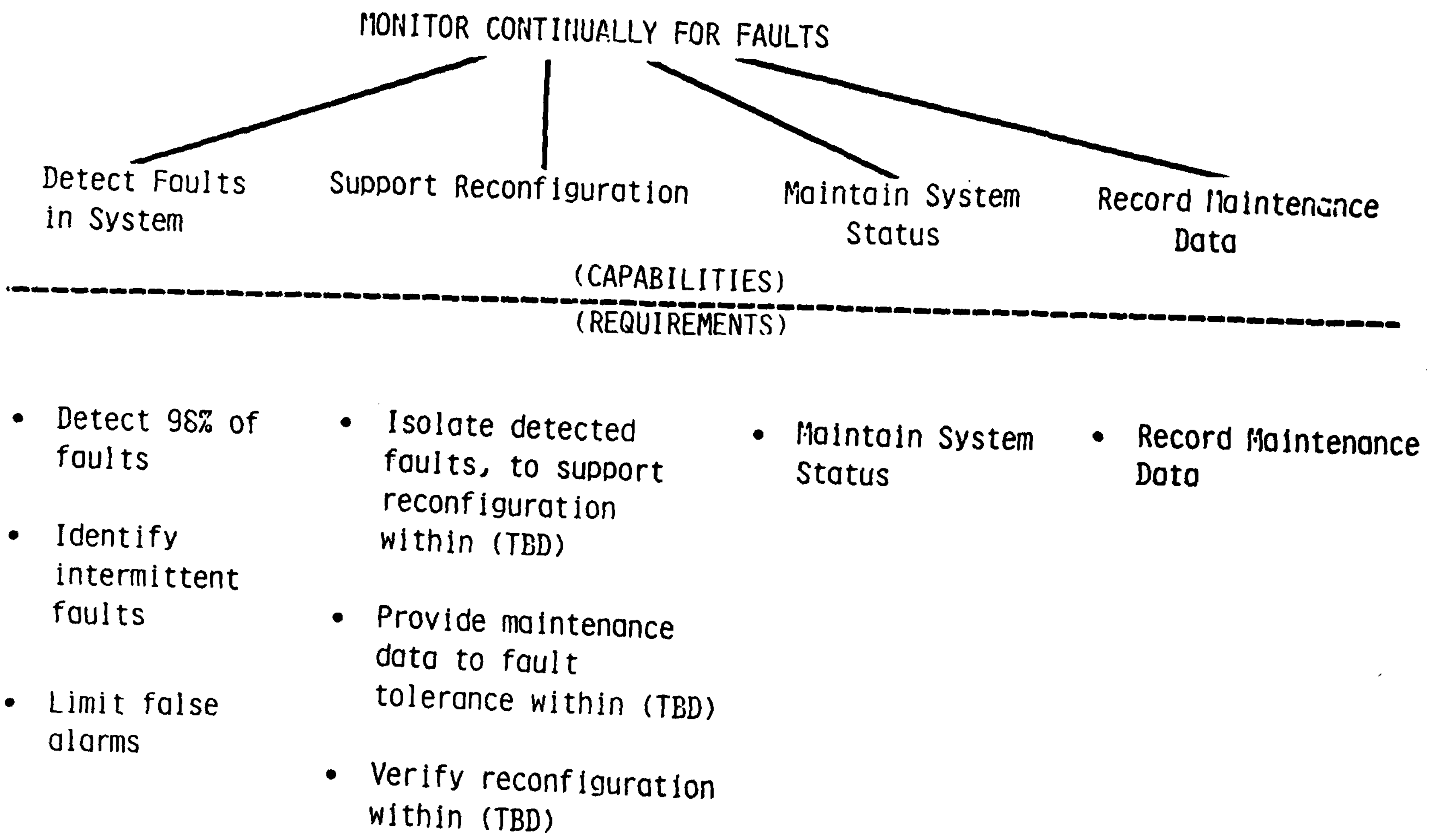
- Detect 98% of faults
- Identify intermittent faults
- Limit false alarms

- Isolate detected faults, to support reconfiguration
- Provide maintenance data to fault tolerance
- Verify reconfiguration

- Maintain System Status

- Record maintenance data

FIGURE 9. Preflight ITM Capabilities/Requirements



MONITOR CONTINUALLY FOR FAULTS

Detect Faults in System

Support Reconfiguration

Maintain System Status

Record Maintenance Data

(CAPABILITIES)

(REQUIREMENTS)

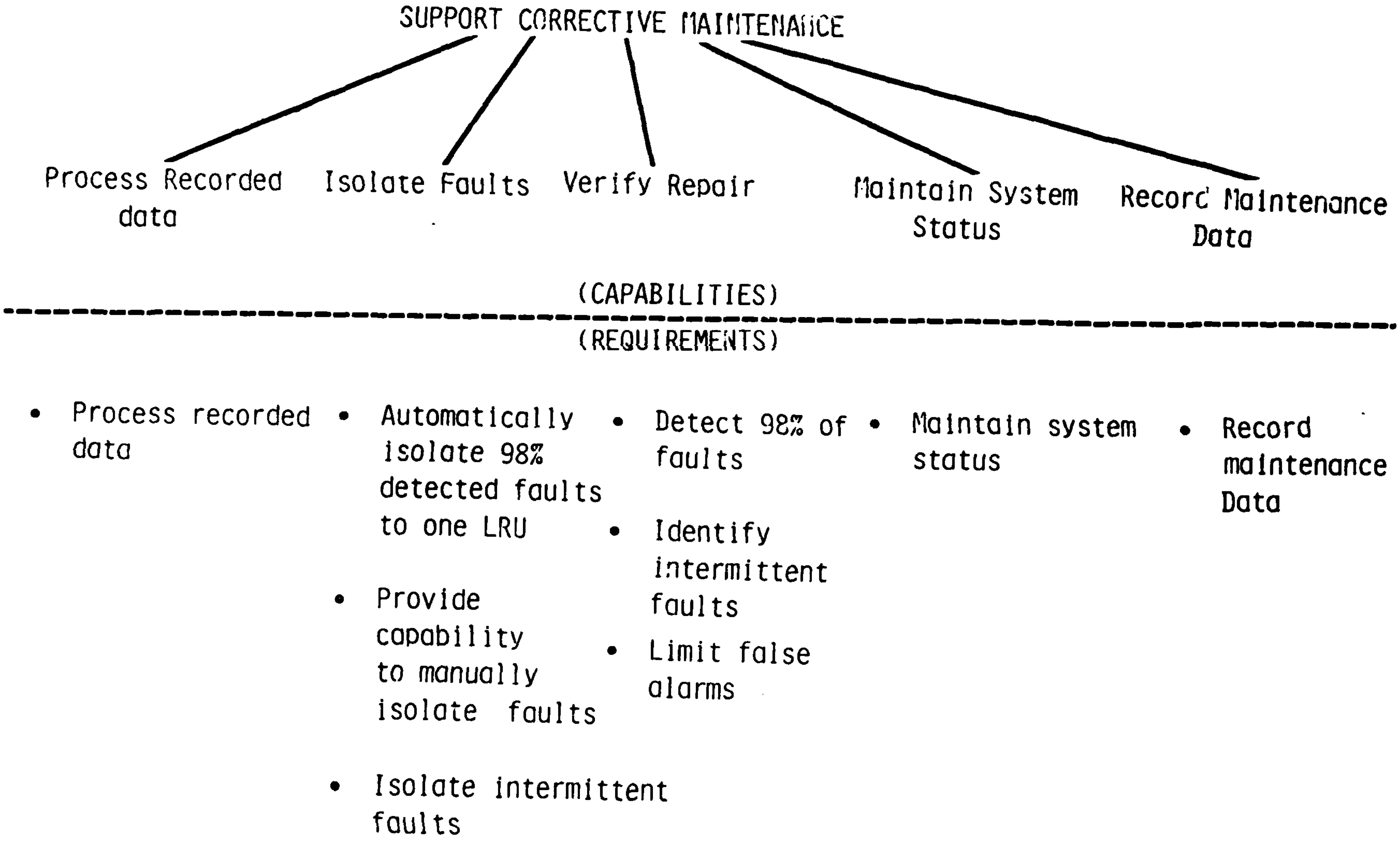
- Detect 98% of faults
- Identify intermittent faults
- Limit false alarms

- Isolate detected faults, to support reconfiguration within (TBD)
- Provide maintenance data to fault tolerance within (TBD)
- Verify reconfiguration within (TBD)

- Maintain System Status

- Record Maintenance Data

FIGURE 10. IN-FLIGHT ITM CAPABILITIES/REQUIREMENTS



SUPPORT CORRECTIVE MAINTENANCE

Process Recorded data

Isolate Faults

Verify Repair

Maintain System Status

Record Maintenance Data

(CAPABILITIES)

(REQUIREMENTS)

- Process recorded data
- Automatically isolate 98% detected faults to one LRU
- Provide capability to manually isolate faults
- Isolate intermittent faults
- Detect 98% of faults
- Identify intermittent faults
- Limit false alarms
- Maintain system status
- Record maintenance Data

FIGURE 11. Postflight ITM Capabilities/Requirements

a.	Prelight	Establish operational readiness
b.	Inflight	Monitor continually for faults
c.	Postflight	Support corrective maintenance

The remaining capabilities and associated system requirements are related as shown. In preflight and inflight, the recording of maintenance data supports the postflight processing.

The ITM system requirements were developed based on three ground rules:

a. The interface to the pilot is minimized. This responds to the concern of pilot workload.

b. No additional ground support equipment is required. This eliminates extra cost and maintenance time burdens.

c. The system should operate on the ground using aircraft power only.

5.0 ANALYSIS OF ADVANCED SYSTEM

The Integrated Testing and Maintenance System (ITM) was developed to be part of the avionic system developed by TRW Defense and Space Systems Group under the Multibus Avionic Architecture Design Study (MAADS) contract, which was awarded at the same time as the ITM contract. The MAADS contract is a first step in defining the architectures and requirements for AFWAL's Advanced System Integration Demonstrations (ASID) designated as the PAVE PILLAR program. The concurrent development under both the ITM and MAADS contracts resulted in ITM's design requirements being based upon an architecture that was periodically being modified. This report is based on the architecture defined at the time the ITM system specification was prepared, which is similar but not identical to the architecture defined in the MAADS specifications.

This section discusses the features of the advanced system relating to the design of ITM and illustrates the architectural and structural configurations upon which ITM is based.

5.1 ADVANCED SYSTEM CONFIGURATION

The advanced system configuration was modified from using four mission processors, each three assigned responsibility for two or more avionics functions and one as a backup, to a design that requires the use of only two mission computers. This was facilitated by the increased computing power provided by computing devices anticipated to be used in the avionics suite. Another major change made in the system configuration concerned the transmission of stores video data. Previously, it was anticipated that the stores video data and the image sensors video data would interface with the image processing subsystem via a video bus. The current configuration provides separate video links from the image sensors and stores to the image processing subsystem.

A major change made to the advanced system architecture was the inclusion of the mission computers on the high-speed bus. Originally, only the fusion processors were connected to the high-speed bus. This required that certain test results be processed by the fusion processors before being transferred to the mission computers. Now, however, those test results may be directly acquired by the mission computers where fault detection and isolation processing occurs.

The advanced system configuration used as a basis for the ITM was developed as part of the MAADS effort and is illustrated in figure 12.

5.2 SYSTEM FUNCTIONAL DIAGRAM

The preceding system configuration was used to develop and guide the concepts for ITM data gathering along with determining the transmission paths of core avionics, fusion processors, subsystem processors, and sensors. The actual functional subsystem design and definition of reasonableness tests, however, is based on the system functional diagram shown in figure 13, also an output of the MAADS effort.

This diagram shows the interrelationship of avionic elements with information paths of the major avionic elements and subsystems. With more detailed diagrams provided by TRW, the makeup of the avionic system started to take place.

Certain areas of the functional diagram currently lack concrete definition. For example, the full complement of target-sensors that are identified in the target acquisition functional-subsystem functional flow, cannot be substantiated at the present time. It was necessary to select sensors that are projected to be included. When the final system structure and avionic elements are defined, the subsystem functional and structural diagrams will have to be modified as well as the reasonableness test and BIT requirements, which are based on them.

5.3 ADVANCED SYSTEMS LRU'S

The composition of the advanced system configuration and functional diagram lead to the identification of the avionic line replaceable units (LRU). This selection was critical since the onboard testing and data recording functions depend on the specific LRU's in the system configuration.

The completment of LRU's assumed for the advanced system was selected using the following methods:

a. Review of the system configuration.

b. Analysis of system functions.

c. Analysis of hardware elements associated with current subsystems.

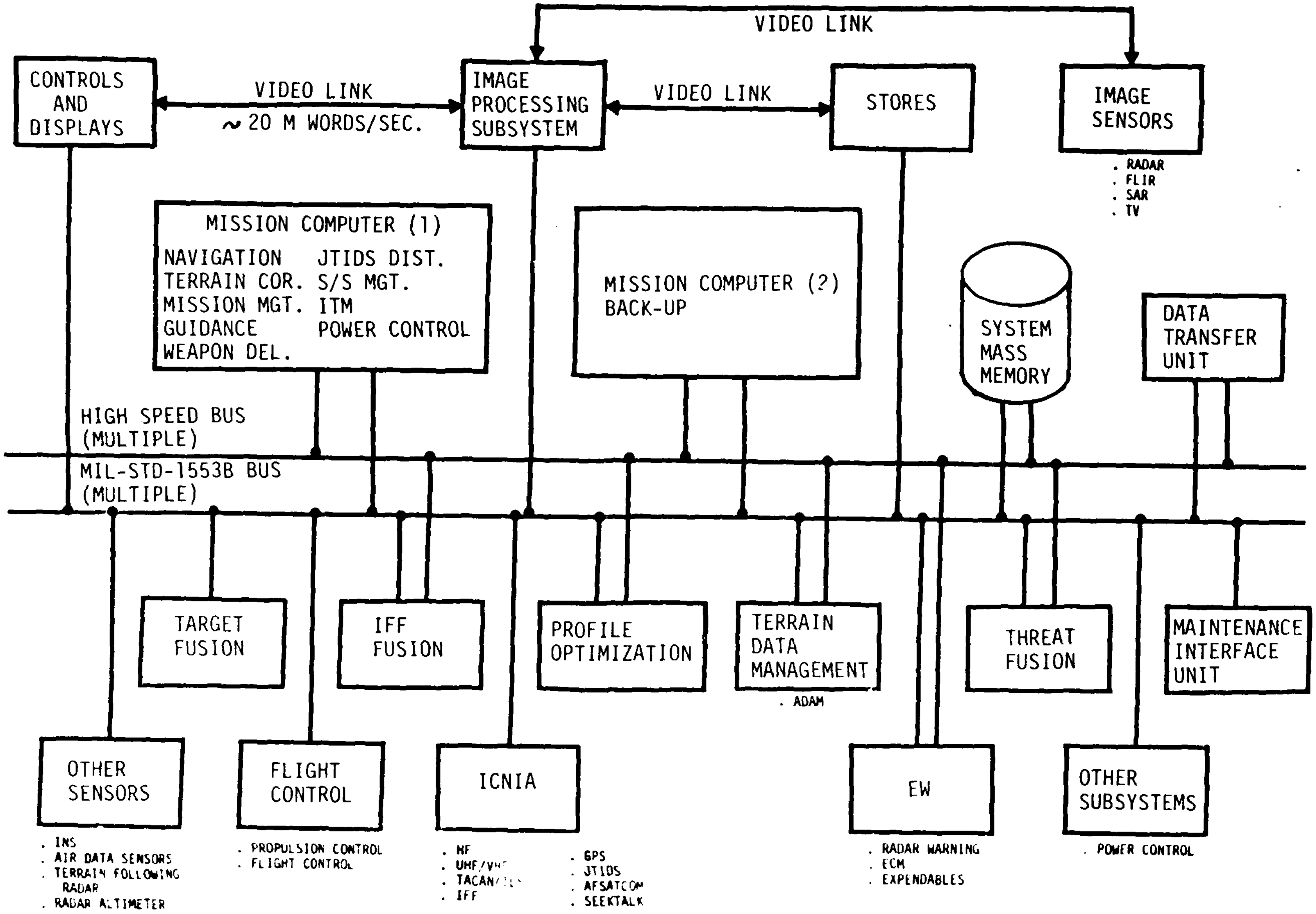


Figure 12. Advanced System Architecture

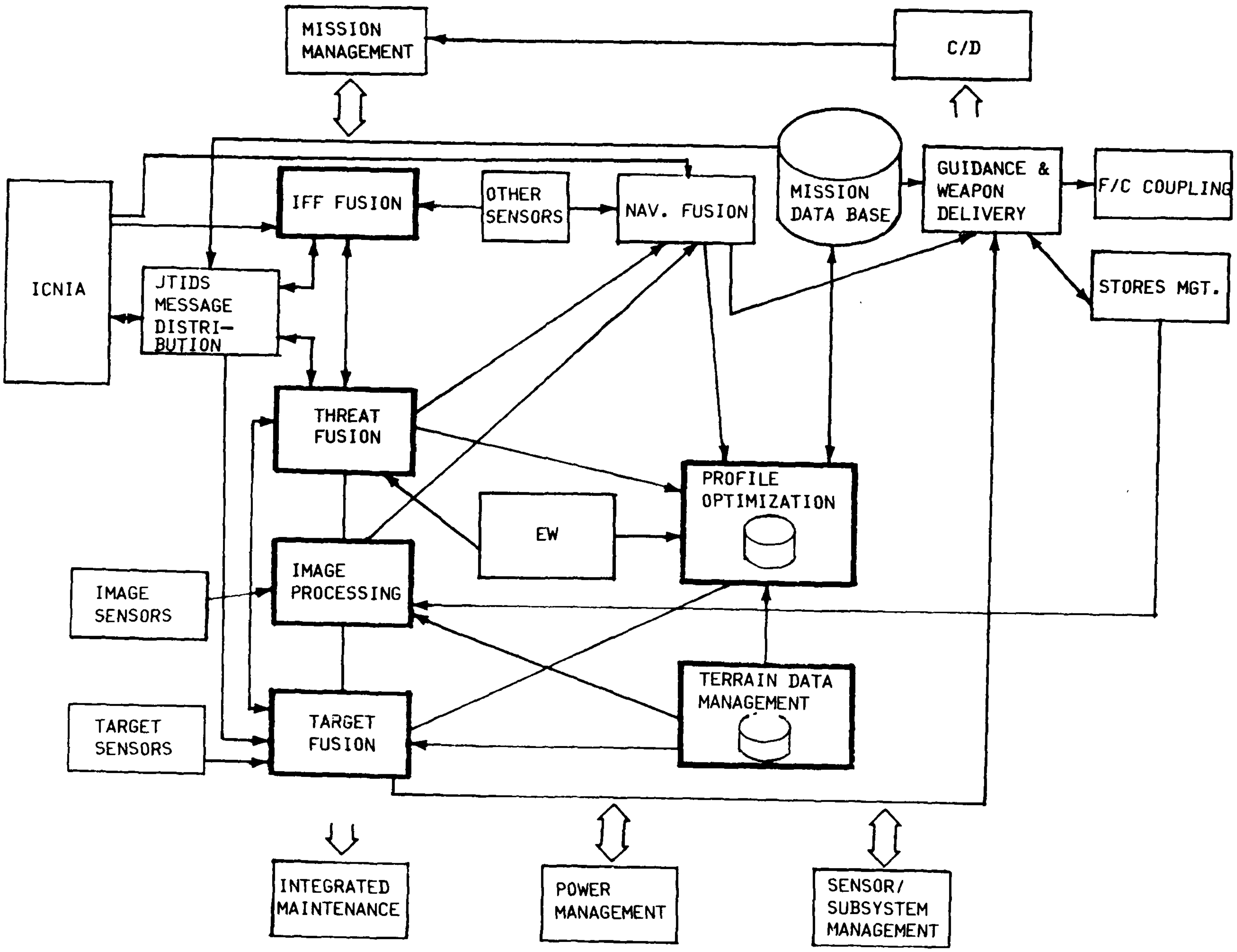


Figure 13. System Functional Diagram

d. Review of mission demonstration specifications for—
   (1) Operational readiness mission
   (2) Survivable penetration mission
   (3) Survivable strike mission

The selected LRU's are listed by structural subsystem in table 9.

5.4 ADVANCED SYSTEM LRU STRUCTURE

The final analysis of the advanced system details the development of an LRU structure using the results obtained from the system configuration, functional descriptions, and the list of avionic LRU's.

The purpose of the LRU structure is to define the data paths and the hierarchy of communication. This is important to ITM development because of the impact on acquisition of built-in test (BIT) data and reasonableness test data.

The developed advanced system LRU structure is shown in figure 14.

5.5 FUNCTIONAL AND STRUCTURAL DELINEATION

It became apparent early in the development of ITM that the increasingly integrated nature of avionic was causing difficulty. Previously, avionic systems were distinct both as functions and as separate hardware groups. Typically each subsystem was self-contained, including its own controls and displays. New implementations of avionics integrated numerous functions using common processors, controls, displays, and sensors. Some ITM functions (e.g., reasonableness testing of mission data) were concerned with the avionic functions. Others (e.g., BIT data acquisition) were concerned with the physical implementation of the avionics. To clarify the distinction, the designations "functional subsystem" and "structural subsystem" were introduced.

Throughout the document functional and structural subsystems will follow the intent of the following definitions:

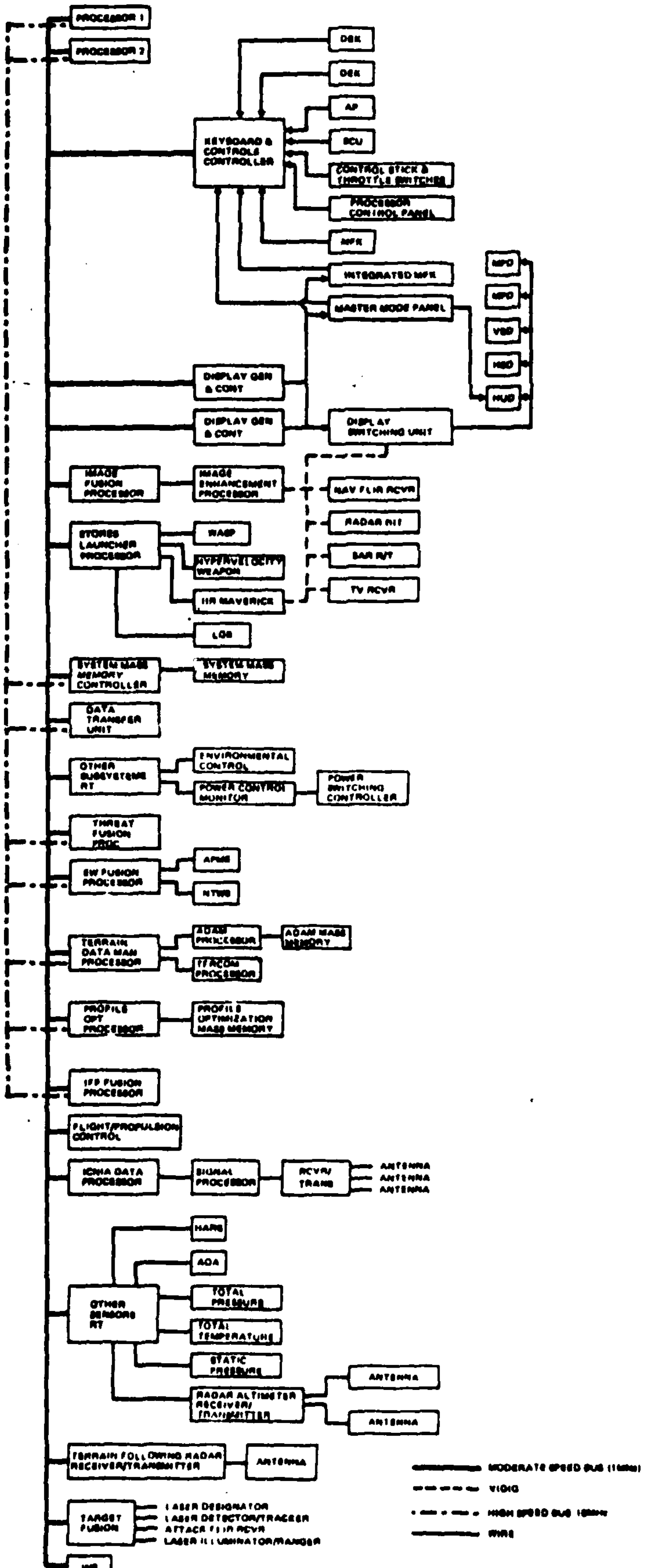
Functional subsystem—A subsystem of the aircraft avionics that performs a function (e.g., navigation, communications). A functional subsystem may integrate several structural subsystems (e.g., navigation uses INS, ICNIA, controls, and displays).

TABLE 9. LIST OF LRU'S

Structural subsystem	LRU
Mission processing	Mission processors (2)
	System mass memory
	System mass memory controller
	Data transfer unit
Controls and displays	Multipurpose display (2)
	Vertical situation display
	Horizontal situation display
	Head up display
	Display switching unit
	Display generation and control (2)
	Master mode panel
	Multifunction keyboard
	Integrated multifunction keyboard
	Data entry keyboard (2)
	Armament panel
	Sensor control unit
	Control stick and thottle switches
	Processor control panel
	Keyboard and controls controller
Fusion processing	Target processor
	IFF processor
	Threat processor
ICNIA	Antenna subsystem
	Receiver-transmitter
	Signal processor
	Data processor
Navigation	Inertial navigation subsystem (INS)
	Altitude, heading, reference system (AHRS)
	Total pressure sensor
	Static pressure sensor
	Total temperature sensor
	Angle of attack sensor
Terrain-following radar	Antenna
	Receiver-transmitter
Radar altimeter	Antenna (2)
	Receiver-transmitter
Terrain data management	Terrain data management processor
	ADAM processor
	ADAM mass memory
	TERCOM processor
Profile optimization	Profile optimization processor
	Profile optimization mass memory

TABLE 9. LIST OF LRU'S (CONTINUED)

Structural subsystem	LRU
Image processing	Image enhancement processor
	Image fusion processor
Image sensors	NAV FLIR receiver
	Radar receiver-transmitter
	SAR receiver-transmitter (uses TF/TA antenna)
	TV receiver
	MMW receiver
EW subsystem	New threat warning system
	APMS
Flight controls	Flight-propulsion controllers (3)
Power control	Power monitor
	Power switching controller
Stores	Launcher processor
	Wasp
	Hypervelocity weapon
	IIR maverick
	LGB
Target sensors	Laser designator
	Laser detector-tracker
	Attack FLIR receiver
	Laser illuminator-ranger



System LRU Structure

Figure 14. Advanced System LRU Structure

Structural subsystem—An end item that is identifiable as a physically distinct item (e.g., INS, ICNIA, radar). Each structural subsystem may support more than one functional subsystem (e.g., ICNIA supports both navigation and communication).

5.6 STRUCTURAL SUBSYSTEMS

There are four primary classes of structural subsystems: (1) core avionics, (2) fusion processors, (3) subsystem processors, and (4) sensors. These categories reflect the system definitions found in the statement of work, advanced mission specifications, and analysis of the advanced system LRU structure. Each physical end item is considered a structural subsystem and in review of the structure of the system, four major divisions encompass the total system.

5.6.1 Core Avionics

Core avionics is the name given to a collective group of structural subsystem end items. The core avionics are those elements of the structural subsystem that are common to all functional subsystems. All core avionics are physically distinguishable items and support one or more functional subsystems.

The core avionics are made up of the following elements:

a. Mission processors.
b. Controls and displays.
c. System mass memory.
d. Data transfer unit.
e. Data buses.

These elements were selected because of their commonality of support to the nine functional subsystems. They also support the data gathering and testing of the remaining structural subsystems. The dual role of these elements ultimately determined their selection.

5.6.2 Fusion Processors

Fusion processors provide result estimation through the processing ,of multiple sources of related data. As an example, the target fusion processor provides a best estimate of the location of the identified target through the merging of target data from such sources as the laser detector and tracker, advanced digital avionics map (ADAM), and synthetic aperature radar (SAR). The effect a loss of a source of data has on the estimated result is determined by the importance placed on the data by the fusion processor algorithm. The fusion processors of the advanced system are: (1) target, (2) threat, (3) identification friend or foe (IFF), (4) EW, (5) image, and (6) profile optimization processors.

5.6.3 Subsystem Processors

Subsystem processors differ in function from fusion processors. Subsystem processors provide computing and processing capability to structurally related end items, which perform similar activities. Similar activity end items transfer raw or processed data to the subsystem processor which performs further processing. The resultant data are then passed down to the original sources of data, up to fusion processors as a related source of data, or up to mission computers that analyze and distribute the data. The subsystem processors of the advanced system are: (1) image enhancement, (2) stores launcher, (3) environmental control, (4) power control, (5) APMS, (6) NTWS, (7) ADAM, (8) TERCOM, and (9) flight and propulsion control processors.

5.6.4 Sensors

The final complement of sensors to be incorporated into the advanced system have yet to be identified. However, references 5 to 7 and the MAADS specification define prospective sensors in addition to a multisensor system. Through review of these references, a complement suite of sensors have been identified as those interfacing with ITM. The identified sensors are described in the ITM system specification.

5.7 FUNCTIONAL SUBSYSTEMS

Nine functional subsystems have been identified by the mission demonstration specifications as outlined in references 5, 6, and 7. These subsystems are: (1) navigation, (2) guidance and flight controls, (3) communications, (4) weapon delivery, (5) target acquisition, (6) stores management, (7) threat management, (8) electrical power management, and (9) mission management.

The development of the content of the above nine functional subsystems, however, was a task performed concurrently with the development of the multibus avionics architecture being undertaken by the TRW Defense and Space Systems Group. As such, limitations were placed on the amount of detail that could be derived from the MAADS architecture and avionic elements.

To support the activities of defining onboard testing for ITM, it was necessary to identify the tasks, structural subsystem items, and information flows for each of the nine functional subsystems. Information was obtained from the Air Force, TRW in its present activity of developing the advanced architecture, Boeing Military Airplane Company in its support of avionics design for the B-1B, Boeing Aerospace Company in its role as prime contractor for the E-3A (AWACS) aircraft, and other Boeing experience in the development of avionic systems.

The content of each functional subsystem is used to identify the reasonableness data to be acquired from and reasonableness tests that are to be performed for each subsystem. System exercises were designed using the structure of the subsystem and the interrelationship of avionic elements. Finally, a more thorough understanding of the concept of fitting ITM to the proposed architecture is obtained when reviewing the subsystem, its modes of operation, its status during flight phases, and the LRU's used to support each subsystem.

Each functional subsystem is illustrated with a functional flow diagram included in appendix B. For those subsystems that include two or more modes of operation, a hardware configuration matrix is included. The matrix identifies the operational data that are provided by a structural subsystem and required by the operating mode of that subsystem.

5.7.1 Navigation

The navigation subsystem included eight operating modes: (1) integrated navigation, (2) global positioning system (GPS, (3) inertial navigation, (4) area navigation, (5) dead reckoning, (6) joint tactical information distribution system (JTIDS) relative navigation, (7) terrain correlation, and (8) position update. As is evident in table B1, integrated navigation provides the greatest navigation detail to support mission operation. As subsequent modes are entered, less navigational capability is provided, resulting in more responsibility being delegated to the pilot, or loss of mission performance.

The ICNIA navigation filter delivers aircraft position, velocity, attitude, acceleration, and heading to the navigation filter after comparisons are made between the ICNIA-derived navigational data and navigational data received from other subsystem items. These items include the INS, air data sensors, terrain following/terrain avoidance (TF/TA) radar, radar altimeter, and update navigation positional data (resident in the navigation state vector). It should be noted that only integrated navigation uses ICNIA supplied information and that it is mixed with other INS navigational data.

A position, velocity, attitude, and acceleration deviation is calculated by the integrated navigation filter and transferred to the ICNIA navigation filter for delivery and updating to the navigation fusion processor.

This navigational data are used by guidance and weapon delivery, profile optimization, ADAM, NTWS, image processing, and threat fusion to support position corrections, terrain correlation, and guidance direction.

Two new navigational aids used within the navigation subsystem are GPS, which is a subfunction of ICNIA (also new), and ADAM. GPS provides 3-D position information from received satellite signals with time and velocity information. ADAM provides terrain profile information from a digital map stored in mass memory. Using the integrated navigation mode of navigation, this map will be constantly updated to give position indications to the pilot. In addition, all detected as well as prebriefed electronic warfare and ground threats will be stored on this map for display.

The navigation subsystem functional flow diagram is illustrated in figure B1.

5.7.2 Guidance and Flight Controls

The guidance and flight control subsystem includes eleven operating modes: (1) integrated guidance, (2) TF/TA, (3) command navigation, (4) 4-D NAV autothrottle, (5) command heading, (6) command track, (7) auto pop-up, (8) TACAN, (9) ILS steering, (10) terrain map, and (11) attack guidance. As with the navigation subsystem, integrated guidance provides the greatest capability to support mission operation with other modes providing reduced performance or application specific guidance. The guidance and flight controls subsystem hardware configuration matrix is shown in table B2.

As is shown in figure B2, the guidance processor is provided with position and velocity data from navigation; terrain hazard data from TF/TA; terrain contour data from TERCOM; target track data from weapon delivery; and route, waypoints, targets coordinates, and threat coordinates from the mission plan resident in the mission database. It uses these data to calculate crosstrack errors, course errors, bank angle limits, IAS, engine pressure ratios, flightpath angles, and pitch commands to be used by the flight control system propulsion control system.

In the integrated mode, steering commands and cockpit displays will be provided that will guide the aircraft in a low-altitude terrain following and terrain-avoidance mission, taking into account the effects of defenses, the impact of terrain to mask those threats, and coordinated attack needs.

5.7.3 Weapon Delivery

The weapon delivery subsystem includes nine modes of operation: (1) integrated fire and flight control (IFFC) auto, (2) IFFC manual, (3) continuously computed impact point (CCIP) auto, (4) CCIP manual, (5) air-to-ground missile (AGM) auto, (6) AGM manual, (7) navigation bomb, (8) Wasp, and (9) hypervelocity. These modes of operation and associated hardware functions are listed in table B3.

In the IFFC auto mode of operation (the most automated) the flight controls are integrated with the fire control system. The weapon delivery computations are based on relative target data from the acquisition sensor, and release is automatic with pilot concurrence. Once the target is acquired, IFFC flies the aircraft automatically so that the aircraft is constantly maneuvering.

The weapon delivery mechanism receives its target data from the target fusion processor. These data are acquired by selecting sensors of the multisensor system, target area map position, and terrain profile from ADAM and target position from JTIDS. The coupling of this information with that from threat fusion, navigation, and stores management allows weapon delivery and guidance to control the aircraft maneuvers necessary for the successful completion of the mission.

Extensive use is made of target data received from MMW, the laser designator, laser illuminator and ranger, and the laser detector and tracker. A description of these target acquisition sensors is included the target acquisition subsystem.

Stores management receives select, arm, fuse, and time-to-go commands from weapon delivery in all modes of operation. However, it is the responsibility of the stores management subsystem to control the release of the requested weapons. As such, weapon delivery selects the type of weapon, arming, weapon spacing, quantity to be released, and simultaneous releases; and stores management approves the release of the weapons.

Figure B3 describes the weapon delivery functional flow.

5.7.4 Target Acquisition

There are two modes of operation—automatic and manual, listed in table B4—in which the pilot can acquire and track targets. In the automatic mode, the target is acquired, classified, and tracked using the multisensor system. This is accomplished by correlating information from all available sensors. Automatic target classification based upon the multisensor cues as well as image classification for the infrared (IR) and millimeter wave (MMW) sensors should dramatically reduce the pilot's workload. In the manual mode, the pilot acquires route points and targets using various cockpit displays. This information can then be handed off to the tracking system.

Illustrated in figure B4. the target fusion processor accepts range to target data from the laser illuminator and ranger, target track data from the laser detector and tracker, target position from JTIDS, target area map from ADAM, target recognition confirmation from the image processor and target and threat locations from computed results derived from the image processor and threat fusion processor. It delivers processed target location information to the profile optimization processor, target acquisition data to weapon delivery, and refined target and threat locations back to the image processor and threat fusion processor for error correlation.

The laser designator, illuminator/ranger, and detector/tracker provide a major contribution in the acquisition of target data. The laser designator is used to illuminate targets for the laser detector/tracker and the delivery of laser guided bombs. The laser detector/tracker detects the reflected laser beam from the illuminated target and automatically tracks the target via the laser spot. Finally, the laser illuminator/ranger provides range information to the designated target.

Another important item to be included in the avionic suite is the synthetic aperture radar (SAR) system. This item provides side-looking radar imagery for both navigation and targeting. Using a phased array antenna on a timesharing basis with TF/TA, SAR will increase the use of space in the radom area.

5.7.5 Communications

The communication subsystem is composed of HF, UHF, and VHF transmission of air-to-air and air-to-ground communications. Flow diagrams for each of the transmission frequencies is shown in figures B5, B6, and B7. As subfunctions of the ICNIA system, they provide antijam communications with secure channels for the pilot.

Descriptions of a few of the ICNIA subfunctions addressing communications are discussed below.

a. SEEKTALK Provides secure, antijam communications in the UHF range for air-to-ground and air-to-air communications.

b. SINCGARS Provides secure, antijam communications in the VHF-FM range primarily for air-to-ground communications.

c. JTIDS Used to send, receive, or share information anywhere in a combat theater. It will provide two-way secure communications, navigation, and identification features. It will also prov 'e tactical information emergency reporting.

d. AFSATCOM A secure, antijam satellite communications network whose primary function will be emergency action-message reporting with a secondary function of status report back.

5.7.6 Stores Management

Stores management, illustrated in figure B8, is a single-mode subsystem whose function includes control over weapon type, arming, weapon spacing, quantity to be released, and simultaneous releases. In its operational mode, stores management receives select, arm, fuse, and time-to-go commands from the weapon delivery subsystem and responds with the status of the selected ordnance.

It is the responsibility of the stores subsystem to manage the configuration and release of weapons resident on the tactical fighter. As such, it initiates, monitors, and terminates the launch sequencing for the laser it uses to determine the launch status of an ordnance is its aircraft power at the store.

In conjunction with the target acquisition and weapon delivery activities performed, the stores management mechanism receives target range and track data which are used to determine the response it provides to weapon delivery for the release of ordnances. Also, target locations received from the multisensor system are combined with those received from the laser acquisition devices.

As a final step prior to weapon release, stores management supplies the sequencing jettison quantity to the launcher processor.

5.7.7 Threat Management

Threat management is a single-mode functional subsystem shown in figure B9. It is the responsibility of threat management to control the threat detection, classification, location, and prioritization process and send the results to the overall avionic system.

Its three primary sources of threat information are the IFF section of ICNIA, which both identifies the threat and provides the threat location as determined by the IFF fusion processor; JTIDS where the threats are identified by subscribers; and the new threat warning system (NTWS), which identifies and classifies threats. An additional source of threat data is received from the mission database. From this source comes a priori threat locations and classifications.

This fusion of threat information is used to determine actions demanded of the pilot. Threat class, location, and priority information is transmitted to cockpit controls and displays as well as to the guidance subsystem to include the profile optimization processor.

Through automatic or manual means, selective or broadband jamming of EW threats may be carried out. Threat avoidance and/or threat destruction measures may be taken after confirmation that the threat location has been jammed.

Finally, threat management in conjunction with image processing allows for unconfirmed threat locations to be evaluated followed by confirmation or denial of threat potential.

5.7.8 Electrical Power Management

Illustrated in figure B10, electrical power management is a single-mode functional subsystem. Its objective is to (1) provide uninterrupted electrical power to the avionics suite, (2) assess avionic reconfiguration measures taken, determine electrical load shifts and modify distribution of power to accomplish the above, (3) control power levels radiated to accomplish selective or broadband jamming of EW threats, and (4) provide power status information to the pilot and maintenance crew for operation and evaluation of avionic and nonavionic activities.

The controlling figure in this subsystem is the electrical power manager. The electrical power manager receives all equipment status information and makes judgmental decisions on continued operation or removal of the faulty item. It also uses this status information in the shifting of power for broadband jamming and the modification of configuration power distribution to ensure proper avionic operation.

An electrical regulator assists in the voltage and frequency control of the electrical power system. Through the closed loop of the power system, power manager, and power regulator, power control and switching are performed automatically.

5.7.9 Mission Management

At the time the ITM system specification was prepared there was no accepted concept for the functions of mission management. A concept was prepared using available data.

With the start of PAVE PILLAR, mission management appears to be taking on the role of an executive over other subsystems within the total avionic system. To this end, an interface must exist with the mission computers, processors, and selected LRU's and sensors. A preliminary functional flow diagram for the mission management subsystem is shown in figure B11.

6.0 DEVELOPMENT OF REQUIREMENTS FOR THE ADVANCED SYSTEM

6.1 SYSTEM OPERATION

The integrated test and maintenance (ITM) system operation concept is to enable the pilot or crewmember to monitor performance of the avionics health while requiring minimum interaction to perform testing and maintenance. The pilot or crew is also able to exercise some control over ITM so that specific tests can be run and specific functions can be performed. The ITM software will be designed so that neither the pilot nor crew are required to wait for its results (e.g., the preflight checkout can be aborted at any time so that inflight or postflight programs can be executed).

6.1.1 Preflight

The preflight system operation concept will involve the pilot or the crew in testing to establish the condition of all line replaceable units (LRU) in the system. ITM will then extrapolate the readiness of functional subystems based on their dependencies on LRU's.

6.1.1.1 Startup

System startup will be via the Processor Control Panel (PCP) (fig. 15). The pilot or maintenance crew operator will powerup the processors with the program switch set to operational test program (OTP). The mission processor will perform power-up test and will signal successful completion of the test to the pilot. The pilot then depresses the LOAD button to load the OTP into core memory and to load the fusion processors with preflight software. The OTP, which includes both the preflight and postflight analysis programs, will then be loaded into the mission computer.

The first action of the OTP will be to establish a communications link with the operator. The operator is then prompted to select the preflight or postflight mode. Selection of preflight mode completes system startup.

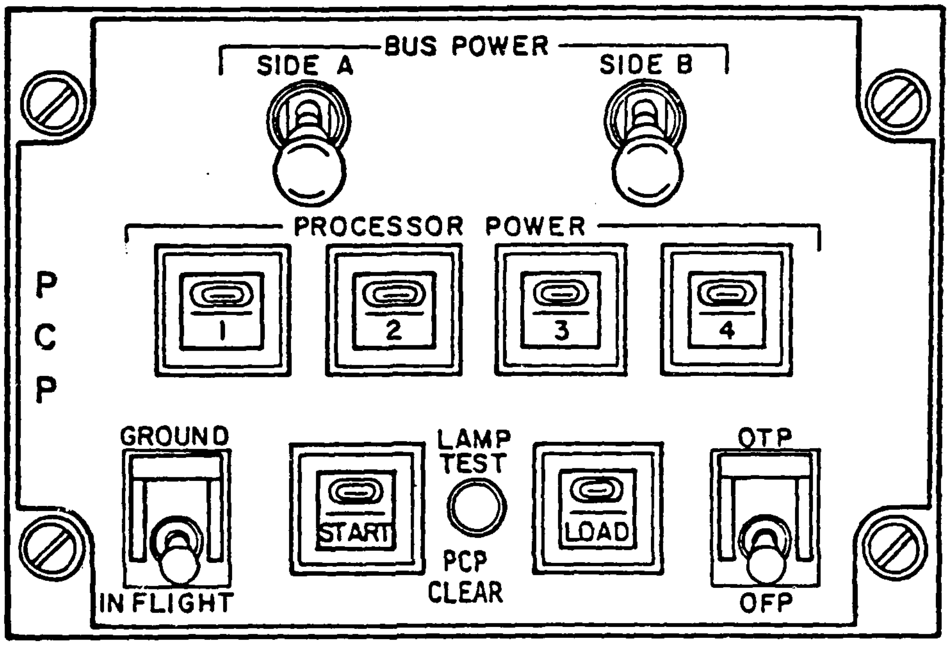
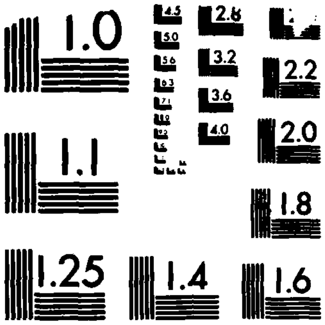


Figure 15. Processor Control Panel

AD-A138 587   INTEGRATED TESTING AND MAINTENANCE TECHNOLOGIES(U)   2/4
BOEING AEROSPACE CO SEATTLE WA R O DENNEY ET AL.
DEC 83 AFWAL-TR-83-1183 F33615-81-C-1517

UNCLASSIFIED   F/G 5/1   NL


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.2
2.0
1.1
1.8
1.25
1.4
1.6

6.1.1.2 Initialization

The system is automatically initialized with configuration data stored on system mass memory. These data are processed before the preflight program begins normal operation.

6.1.1.3 Checkout and Testing

The system performs checkout and testing in a heirarchial manner beginning with the master processor and proceeding outward through the system topography. An estimate of the preflight timeline, described in the software structure section, is shown in table 10.

The pilot or crewmember must respond to system cueing for checkout of the controls, displays, and voice communications. Typically, the system will display test patterns on each CRT for checkout by the crewmember. Also, each keyboard and control device will be exercised by the crewmember. The sequence of actions will be controlled by the ITM system and will be simple enough to understand with little training.

6.1.1.4 System Status

After preflight checkout is complete, the go/no-go status of the system is displayed on the multipurpose display. If there are any failures in the system, they can be reviewed by cueing one of the status displays. They are—

a. Subsystem status display—a single display that shows the status of each of the functions (e.g., terrain following/terrain avoidance (TF/TA), navigation (NAV) bomb, laser-guided bomb). This display will be used by the pilot to assess system effectiveness.

b. Equipment status display—a single display that shows the status of each hardware structure (e.g., integrated communication, navigation, and identification avionics (ICNIA), radar altimeter, stores launcher processor). The pilot can access this display if he desires more detailed breakdown of system status.

TABLE 10. PREFLIGHT TIMELINE (NO FAULT)

Action	Time (sec)	Cumulative time (sec)
Power up avionics	30	30
Wait for warmup and self-test	30	60
Select OTP and initiate startup	10	70
Program load	5	75
Verify communication with processors and displays	5	80
Display message and select mode	5	85
Verify communication with all devices	5	90
Display message and select mission type	10	100
Perform initial BIT check and setup exercise routines	5	105
INS alignment	(90 to 720)	(N/A)
Cyclic monitoring and controls, displays, and communications checkout	300	405
Display system status and end-of-preflight message	5	410

Total 6 min 50 sec*

*Less INS alignment.

c. LRU status displays—a set of displays containing go/no-go status of all LR'Us in the system (e.g., mission processors, angle-of-attack sensor, radar receiver-transmitter). This display is used primarily by the maintenance crew.

6.1.1.5 Shutdown

The pilot or crewmember terminates the preflight operation by shutting off processor power at the PCP. Alternatively, the user can switch to the postflight program if isolation to the LRU level is desired.

6.1.2 Inflight

The goal of ITM during flight is to provide continued monitoring and filtering of faults while minimizing pilot interaction. Data from intermittent and transient faults are stored on mass memory for analysis during postflight phase. Hard faults are detected and isolated so that fault tolerance actions can be taken.

6.1.2.1 Startup

The Operational Flight Program (OFP) is started by setting PCP switches to INFLIGHT and OFP, applying power to the processor, and depressing the LOAD button.

6.1.2.2 Initialization

The inflight test program (IFTP) is initialized with configuration and architecture data and results of preflight testing stored on mass memory. Inflight initialization does not require any pilot interaction.

6.1.2.3 System Status

The pilot can request system status displays (as in preflight) at any time during flight. Built-in test (BIT) test failures and uncritical fault indications are not presented to the pilot unless they infer a flight critical or mission critical problem.

6.1.2.4 Self-Test Initiation

The pilot can initiate self-test on equipment. A menu is presented on the Multipurpose Display (MPD) and selection is made through the Data Entry Keyboard or the MPD. Devices are normally taken offline for initiated testing; so some devices (e.g., INS, HARS) may not support interruptive testing during flight.

6.1.2.5 Suspicion Identification

The pilot can notify ITM of suspect or failed boxes through the suspicion identification command. ITM will respond by supplying the pilot with current test results (including BIT and reasonableness) for the suspect unit. Pilot may cue fault tolerance actions independently of test results by follow-on actions.

6.1.2.6 Shutdown

Shutdown of the OFP is performed by switching off power at the PCP.

6.1.3 Postflight

The objective of postflight operations is to identify and support corrective maintenance actions. Fault data that were collected during preflight and inflight phases is analyzed and supplemented with diagnostic testing. Isolation to the LRU level allows crewmembers to remove and replace a single box for corrective maintenance. A timeline for the postflight procedure is in table 11.

The cockpit procedures are designed so that a pilot with little or no training can perform them if the plane is operating in a sealed cockpit scenario.

6.1.3.1 Startup

Startup of the postflight program is the same as for preflight, except the pilot or crewmember selects postflight portion of the OTP when cued on the MPD.

TABLE 11. POSTFLIGHT TIMELINE

Action	Time (sec)	Cumulative time (sec)
Power-up avionics	30	30
Wait for warmup and self-test	30	60
Select OTP and initiate startup	10	70
Program load	5	75
Verify communications with processors and displays	5	80
Display message and select mode	5	85
Verify communication with all devices	5	90
Display message and select mission type	10	100
Perform initial BIT check and correlate faults to inflight detected faults	60	160
Display message and prioritize fault list	120	280
Automatic fault processing	(780/fault)	(N/A)
Manual fault processing	(1110/fault)[a]	(N/A)
Display intermittent data and select faults to be isolated	60	340
Intermittent fault processing	(160/fault)	(N/A)
Format and display maintenance report data	300	640
Prepare maintenance report and transfer report to DTU, transfer unresolved fault data to DTU	600	1240
Display end of postflight message	5	1245

Total 20 min 45 sec[b]

[a] May be longer.

[b] Less any fault processing.

6.1.3.2 Initialization

The postflight program is initialized with configuration and architecture data from mass memory.

6.1.3.3 Diagnostic Procedures

The diagnostic procedures begin with an analysis and classification of faults that have occurred during and before flight. These faults are classified by symptoms since no detailed isolation has yet taken place. The fault list is displayed on the MPD to allow the pilot or crewmember to reprioritize the isolation order. Otherwise, isolation is done by a preassigned criticality assessment (i.e., hard, intermittent, transient, recovered).

The fault isolation process uses the full implementation of the fault handling scheme described in section 6.3.5. Hard faults are isolated to the LRU-level diagnostic tree analysis. Intermittent or transient faults are isolated by correlation of fault occurrence with mission phase, device modes, environmental data, other faults, and time to find common elements, which will aid the crewmember in deduction of fault causes.

6.1.3.4 System Status

When isolation is complete, the crewmember can access the LRU status displays (described earlier) to determine which LRU's should be replaced. Additionally, he can access new test data gathered in flight and rerun tests. The visibility of test data and results gives the crew member on-the-job training and allows human analysis and reasoning to aid flightline maintenance.

6.1.3.5 Data Off-Load

ITM maintains an interface with the DTU so that offline processing facilities can be used for fault analysis, maintenance history, trend analysis, intermittent fault identification, determination of false alarms, and reproduction of flight environment for simulation pruposes. The data that are off-loaded (shown in table 12) are transferred from mass memory to the DTU under control of the ITM system.

TABLE 12. DATA REQUIRED FOR OFFLINE PROCESSING

Master data

- System configuration
- System status

Fault data for each fault detected

- Aircraft environmental data (LRU temperature, flight dynamics)
- Time
- Reconfiguration measures taken (if any)
- System status
- Fault type
- Fault isolation measures taken
- Fault location (to LRU when possible)
- Test data
- Tests conducted

Number, frequency, and duration thresholds for intermittent faults

Flight phase changes

Subsystem mode changes

6.1.3.6 Maintenance Interface Unit

The MIU is the single piece of ITM-dedicated hardware in the avionic system. Located in the nose wheel well, it is used to display LRU data to the crew at repair time. Messages are scrolled over the 16-character display when cued by the operator; the operator can also review or rerun the message sequence. Data displayed on the MIU include identification of which LRU's need to be replaced, any nonavionic items that require maintenace, and any consumables that need maintenance.

The MIU must be capable of operating under its own power when avionics electrical data bus is shut down for repair, since power and data cables cannot be disconnected while power is applied.

Although the postflight analysis cannot be controlled from the MIU, a crewmember can climb down from the cockpit and make all repairs without returning. Also, a pilot-crewmember team can perform all maintenance in quick turnaround or hazardous environment (sealed cockpit) scenarios.

6.1.3.7 Mission-to-Mission Data Catalog

The last action of the postflight program is to compress all collected fault data and purge unneeded data from mass memory. The mission-to-mission fault data catalog is the compressed fault results over the operating life of the airplane. Included for each fault type are MTBF and maximum frequency rates. These data serve as a comparison that can be used by the crew member or by the test threshold designer to evaluate the test parameters, LRU performance, and system performance.

6.1.3.8 Shutdown

The postflight program is shut down by switching off power at the PCP.

6.1.4 Data Recording

Time and processing limitations during the preflight and inflight phases restrict the amount of time that can be spent on ITM functions. Data are recorded during those phases so that ITM can perform those functions (e.g., isolation to the LRU) during the postflight phase. Basically, two types of data are needed: (1) data that detail the nature of the fault, and (2) data that detail the nature of the operating scenario in which the fault occurred. The former is recorded whenever a fault occurs; the latter is recorded whenever a feature of the scenario changes.

6.2.3.1 Environmental Data Recording

Environmental stress contributes to errors and inconsistencies in BIT systems. A trade study was performed to determine if benefits from recording these data during flight outweigh the costs of implementation and operation. Conclusions from the study were that LRU internal temperatures, translational acceleration, and angular acceleration should be measured and recorded when faults occur and that ambient temperature, system noise, vibration, external power transients, EMI/RFI, contamination, water intrusion, and shock should not be recorded.

Recording of environment data is useful in processing intermittent fault data. When intermittent faults are caused by changes in environmental conditions, having the environmental data when an intermittent fault occurs will help understand and isolate the problem. For instance, if a radar system is being subjected to high (out of specification) noise levels from outside the system, erratic behavior may cause the radar's own BIT or system-level tests to indicate a failure when the box itself has performed within specifications. Recording the system noise level at time of test failure allows ITM and/or the ground crew to correlate the environmental stress with the failure and avoid (1) unnecessary removal, (2) an unexplained cannot duplicate (CND), and (3) dissatisfaction with the BIT.

Environmental data recording can also aid in pinpointing a box's inability to perform up to specified stress levels. Postfight correlation of test results and temperature data, for example, might show that the failures coincide with a certain box experiencing abnormally high (but not out-of-specification) temperatures. Here, the fault isolation capability of the system increases due to the recording capability.

Also, during inflight processing, fault isolation procedures are aided in breaking ties in voting and comparison tests by examining which box was suffering environmental stress. Of course, this is possible only for data types that were LRU independent (e.g., internal temperature)—not for generic system data (e.g., g forces).

Costs of recording environmental data stem from implementation of environmental measuring hardware and from use of system resources to collect and process the environmental data. The method of data handling and hardware implementation were estimated for each of the proposed environmental data types. Implementation is shown in table 13.

While costs of data collection resources are easily defined in terms of bus and processing load, measurement hardware implementation costs have many contributing factors. Design requirements must determine space necessary for the hardware. Retrofits may be necessary for standard equipment. Sensor technology development may be necessary for some types of data. Each data type must be assessed individually to determine the extent of implementation costs.

A cost benefit comparison for environmental data recording is shown in table 14. All scoring catgories have been weighted equally. Each data type was rated on a zero through five scale in each scoring cateogry. Overall comparison numbers were generated by subtracting costs from benefits. Scoring criteria for the categories are as follows:

a. Additional correct isolation. High score resulted from a high independence of stress conditions between LRU's.

b. Reduction in false alarm removal. High score resulted from the likelihood of transient faults caused by the stress, because recording this parameter would allow isolation of the cause of the transient. Low score resulted from low-fault likelihood or if the faults caused by the stress are unrecoverable, hence there is no need to record the data.

c. Cost of hardware implementation. High score resulted from requirement for LRU retrofit, design modifications, or separate hardware. Low score if data are already available.

d. Data handling resources. High score resulted from extensive bus loading or processor computations.

TABLE 13. ENVIRONMENTAL DATA RECORDING IMPLEMENTATION

Data type	Data requirement	Hardware implementation
Internal temperature	one value per LRU	sensor in each LRU
Ambient temperature	severàl values per system	sensors in some LRU's
System noise	one value per system	sensor in single LRU
Translational acceleration	one value per system	no additional hardware
Angular acceleration	one value per system	no additional hardware
Vibration	one value per LRU	sensor in each LRU
External power transients	several values per system	sensors in some LRU's
EMI/RFI	one value per LRU	sensor in each LRU
Contamination	one value per LRU	sensor in each LRU
Water intrusion	one value per LRU	sensor in each LRU
Shock	one value per LRU	sensor in each LRU

TABLE 14. ENVIRONMENTAL DATA SCORING

Data type	Additional correct isolation	Reduction in false alarm removals	Cost of hardware implementation	Data handling costs	Overall
Internal temperatures	5	5	2	2	6
Ambient temperature	2	4	3	2	1
System noise	4	3	4	3	0
Translational acceleration	3	3	0	1	5
Angular acceleration	3	2	0	1	4
Vibration	4	3	4	3	0
External power transients	3	2	4	1	0
EMI/RFI	5	4	5	3	1
Contamination	2	0	5	1	-4
Water intrusion	1	0	4	1	-4
Shock	1	2	3	1	-1

The ITM approach will be to record those variables that appear to provide relatively good benefits without incurring high costs. Internal temperatures, translational acceleration, and rotational acceleration were those selected for fighter avionic systems.

6.1.4.2 Failure Data

In addition to environmental data, fault data recorded upon failure of built-in tests, reasonableness tests, and statistical tests are recorded on the system mass memory. These data are designed to support correlation of system conditions during postflight so that inferences can be drawn either by the ITM system or by the crewmembers.

6.1.4.3 Scenario Data

Operational scenario data are also recorded on system mass memory. These data include operating modes of subsystems or LRU's, system status, system configuration, and flight phase. Whenever any one of these variables changes, the change along with the mission elapsed time is recorded. The postflight processing software will reconstruct the scenario at the time of any fault occurrence.

6.2 INTERFACES

It is the purpose of ITM to receive information from and provide information to avionic subsystems and to receive information from nonavionic subsystems. To accomplish this, information transfer is conducted through external and internal interfaces to ITM.

Interface development and definition was approached differently for the external and internal categories. External interfaces address functional and structural elements that provide data to ITM to support ITM-driven testing routines. As such, a major portion closely followed the identification of system exercises and reasonableness tests for functional subsystems and BIT for structural subsystems.

Internal interfaces are primarily those software module interfaces within ITM that direct the transfer of data between flight phases and include fault data, system status, and system configuration. These software modules are identified as the Operational Test Program (OTP) that contains separate preflight and postflight modes and the Inflight Test

Program (IFTP) that is to be included as part of the OFP. In addition, the MIU hardware element that provides an interface to the ground crew during remove and replace and service activity is defined as an internal ITM interface.

6.2.1 External Interfaces

As ITM is predominantly an onboard test system used to detect and isolate failures in avionic and nonavionic systems, its scope is felt throughout the avionic and nonavionic suite. To support this broad objective, an interface is required between the ITM and most activities resident within the aircraft. ITM acquires test results, operational data, and operating procedures from both avionic and nonavionic elements throughout the aircraft.

To ease the integration of ITM with all elements of the aircraft, each activity or process was categorized according to how it interfaces with ITM and what data or control it provides. In addition, the processes within ITM were identified and correlated to an external activity. The result was a flow of information and control, which began with the acquisition of failure data and concluded with offloading of that data to perform postflight analysis.

All aircraft activities are listed under six external interface categories. These are—

a. Structural subsystems.
b. Functional subsystems.
c. Nonavionics.
d. Pilot-crew.
e. Offline processing.
f. Fault tolerance.

Each category provides unique parameters of operations to be used for the performance or control of ITM failure detection and isolation. These parameters or operations, in turn, support specific tasks performed within ITM, which is the core of ITM activity.

As an example, structural subsystems are identified as a unique class of external interface. By definition, structural subsystems are end items that are identifiable as physically distinct items (e.g., INS, ICNIA, radar). Each structural subsystem may support more than one functional subsystem (e.g., ICNIA supports navigation and communication). A structural subsystem provides test results to ITM for noninteruptive and initiated BIT. This includes data such as LRU temperature, voltage levels, or data bus tests. The task within ITM that the structural subsystem supports is BIT data collection.

This process was continued through all aircraft activities. An illustration of all external interfaces and the tasks within ITM they support is shown in figure 16.

6.2.1.1 Pilot and Crew

The pilot and maintenance crew conduct test control and status display activities interactively with ITM during the preflight, inflight, and postflight phases of mission operation. To facilitate this in an orderly and effective manner, guidelines and procedures have been established. These guidelines and procedures are categorized into the following groups—

a. Display guidelines.
b. Error and input control.
c. Preflight operations.
d. Inflight operations.
e. Postflight operations.

Display guidelines—The purpose of display guidelines is to ensure that all messages, failure notifications, and pilot-crew cues are easily understood and clearly presented. However, another important consideration is the thorough presentation of display information made available to the pilot or crew. To accomplish this, the following types of display information necessary for ITM input, control, and operation have been identified.

Keyboard entry data
BIT results
Data bus status
Critical system notice
Redundant system availability
Invalid command
Test pattern
Hard failure notice

Menus
Individual avionics status
Failed system-subsystem notification
Out-of-tolerance notification
Redundant subsystem availability
Pilot-crew cues
Intermittent fault notice
Equipment recovery

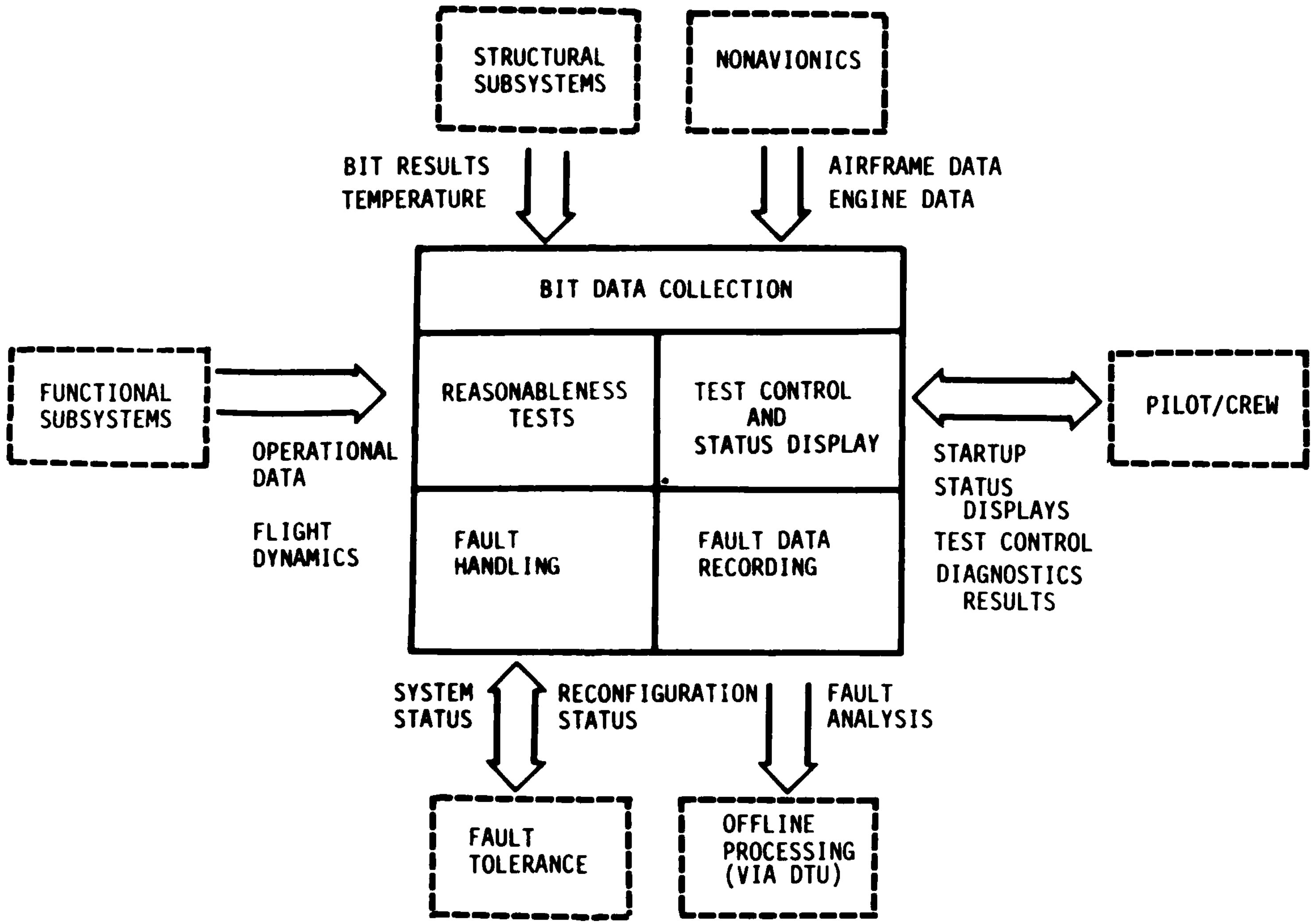


Figure 16. ITM External Interface

Intermittent fault data

Initialization complete

Hard fault data

Avionics maintenance priorities

A second critical area to be identified is the menu items used to perform initiated BIT's. This had to be closely aligned with the BIT and reasonableness tests to ensure that all functional and structural processes within the avionic suite fall within the boundaries established for each initiated test. The categories of initiated tests and their associated menu items were developed using the advanced system LRU structure (see fig. 18). All structural end items are identified and grouped according to major activity. The defined initiated test menu items are shown below.

System mass memory

Inertial navigation system (INS)

Power control

TF/TA

TERCOM

Image sensors/processing

Attitude and heading reference system (AHRS)

Displays and controls

System processors

ICNIA

Air data sensors

Stores management

Advanced digital avionics map (ADAM)

Flight/propulsion control system

Target sensors/processing

Electronic warfare (EW)

System buses

All

Error and input control—Error and input control guidelines are used to optimize compatibility with the pilot and crew and to minimize other factors that degrade human performance or contribute to human errors. To be consistent with current military systems and equipment and to ease the merging of ITM into avionic systems, standard military guidelines were followed in specifying the control procedures. Those procedures can be found in MIL-STD-1472C.

Preflight, inflight, and postflight operations—The operations in the three flight phases define system startup and initialization procedures, outline the loading of the ITM software modules, and discuss offline processing of data collected during flight by ITM. These operations are discussed in detail in section 6.1.

6.2.1.2 Offline Processing

Offline processing is the name given to fault analysis and other postflight activities performed in the nonaircraft environment. ITM acquired and processed data is downloaded onto the DTU and transferred to the desired medium for storage and subsequent processing.

Recently, with the advancements in speed, capacity, and cost in volatile and nonvolatile memory, increased emphasis has been placed on the recording of aircraft operational data along with typically nonavionic data to support more thorough fault analysis and trend history. This was necessitated by the increased cost of avionic systems along with the increased complexity of avionic and nonavionic systems.

With offline processing being of such large scope, it was necessary to identify which activities offline processing will support. This then drives the type of data and amount of data that will subsequently be required to be recorded.

Through review of efforts undertaken by Boeing and others in the BIT community, and especially comments received from the military, it was concluded that increased amounts of mission, operational, and environmental data were necessary. A quick look back to the Currently Achieved Testing Results section of this document illustrates that intermittent faults, CND's, and retest OK's (RTOK) are at unacceptable levels. With this in mind, six activities to which offline processing will be directed are briefly discussed below.

6.2.1.2.1 Fault Analysis

This major first task addresses the identification and determination of failure type. With this analysis resides the location of the failure, the time, its impact, avionic and nonavionic conditions at the time of the failure, isolation measures that were taken to remedy the identified problem, reconfiguration measures taken (if any), and the tests that were conducted to detect and isolate the failure. All subsequent activities listed in some way support the tasks addressed above.

6.2.1.2.2 Maintenance History

Increased emphasis on recording failure data leads to the tracking of failures resident in specific classes of LRU through the documentation of each failure and test data. A maintenance history by LRU assists the intermediate level and depot-level personnel in removing historically deficient boxes and aids the maintenance technician in system troubleshooting by providing a basis to start this inspection.

CND's can also be greatly reduced by providing the organizational level maintenance technician with a history of events that led to the failure. By reviewing the parametric data existing at the time of the failure, the technician may better reproduce the operational environment and analyze the results. This will lead to a reduction in CND's since this snapshot of events will allow a visible history of the failure to be walked through.

6.2.1.2.3 Failure/Trend Analysis

With the above maintenance history integrated into postprocessing routines and with fault analysis results documented, failure and trend analysis can be accomplished to assist in the projection of faulty boxes and the removal of these boxes before system degradation can occur.

Carnegie Mellon University has done some preliminary work on trend analysis concerning interarriving rates of false alarms. Deterioration of system performance is indicated when interarrival times decrease. Prediction of times for removal of units is therefore possible, prior to any detection of such degradation by the pilot.(3) These and other studies show an increased interest and technological support for trend analysis.

6.2.1.2.4 Intermittent Fault Identification

It is widely concluded that the subject of intermittent faults is the most troublesome area encountered by BIT and the maintenance community. In order to determine the seriousness of the intermittent problem, and in many cases reproduce it, postflight processing is critical along with operational and environmental data.

Studies addressing intermittent faults point to data recording as an assistance in helping overcome some of the current deficiencies. In order to better identify intermittent faults, increased memory requirements are placed on the system memory (system mass memory) and new memory requirements are placed on LRU's.

The use of this nonvolatile memory and subsequent memory inspection as part of the diagnostic capabilities should aid the maintenance technician in system troubleshooting, particularly in identifying intermittents and environmentally related failures (to be discussed later).

6.2.1.2.5 Determination of False Alarms

Two categories of false alarms were identified as part of this study. Category I false alarms are those indicated faults where no fault exists (per MIL-STD-1309B). Alarms of this category are caused by false indications due to design errors in the test system or tested unit. Category II false alarms are those indications of a failure when no maintenance is required. The equipment fails to meet specified performance due to external conditions (e.g., temperature, noise) but returns to specified performance once the cause is eliminated.

These fault indications can have an adverse effect on the normal operation of the mission aircraft and therefore must be analyzed and categorized during postprocessing.

6.2.1.2.6 Reproduction of Flight Environment

A typical problem encountered by intermediate-level (where required) and depot-level personnel is the inability to reproduce the flight environment on the ground. For example, a pitch computer may be unnecessarily replaced when the only problem encountered is 'porpoising' in the pitch axis due to adverse wind conditions.

The reproduction of the flight environment will lead to a stronger replication of the flight conditions and further the success of duplicating failures and isolating defective LRU's, SRU's, and components.

Support of the activities listed above requires the recording of a considerable amount of mission operational and environmental data. After researching the practices that are currently applied and what advances are advocated, we have identified categories of fault data that must be recorded. Table 8, shown previously, lists the data to be recorded each time a fault is detected to adequately support evaluation of faults at the maintenance shop.

6.2.1.3 Fault Tolerance

Fault tolerance implies the absence of any single-point failures, and automatic reconfiguration in case of a failure. As such, fault tolerance determines the system capability affected by a fault and commands an equivalent or degraded system configuration from the resources remaining.

By contractual definition, ITM is not to develop the fault tolerance function, but is to support fault tolerance through exchanging pertinent parametric information. To accomplish this, a handshaking approach between ITM and the fault tolerance system being developed by the TRW Space Systems Group must be specified to ensure desired results.

The responsibility placed upon the two functions differs according to the purpose of the system. With the fault tolerance system required to reconfigure the failed or faulty subsystem, it is the responsibility of ITM to provide adequate current state and fault data to allow fault tolerance to determine a suitable equivalent or degraded capability. As with the responsibility placed upon ITM to supply current status information, the fault tolerance system should inform ITM of the reconfiguration measures taken in order for ITM to continue with its tasks and update its status data in support of further avionic activities.

To accomplish the above, ITM must provide the following parametric data to the fault tolerance system:

a. Current system status.
b. Current system configuration.
c. Current configuration resources.
d. Type of fault.
e. Fault isolation results.

In return, the fault tolerance system is required to provide the following resultant data:

a. Reconfiguration status.
b. Reconfigurated system configuration.

6.2.1.4 Functional Subsystem

For purposes of describing ITM's functional subsystem interfaces, the avionic system followed the functional subsystem modes as described in section 5. The functional subsystems are navigation, guidance/flight controls, communications, weapon delivery, target acquisition, stores management, threat management, electrical power management, and mission management.

The functional subsystem interfaces were chosen to support reasonableness testing and other statistical tests in both the preflight and inflight phases of mission operation. These reasonableness and other statistical tests are discussed in section 6.3 under testing approaches.

If no reasonableness test is defined for a function during a certain phase of aircraft operation, there are no external interfaces identified in support of that phase. This however, does not imply that no onboard testing is being conducted. As discussed in section 5.5, a functional subsystem integrates functions of several structural subsystems and as such, initiated and noninterruptive BIT tests are performed on the structural subsystem elements of each functional subsystem.

The flight phased data provided to ITM by each functional subsystem in support of reasonableness and other statistical tests can be found in the ITM system specification.

6.2.1.5 Structural Subsystem

The external interfaces for each of the structural systems are required to collect BIT results. The BIT data include the results of environmental tests (e.g., temperature), voltage-level tests, computer diagnostic tests (e.g., interface echo, address data bus, and tests specific to the application being addressed. This final category is aimed primarily at the sensors since imaging RF and infrared detection demand very specific analysis.

The selection of the test result for each structural subsystem is based on the matching of system complexity, element criticality, and the amount of upward or downward effect a failure has on other elements. As an example, the controls and displays (part of core avionic) is coupled with almost every avionic and nonavionic subsystem throughout the aircraft. Therefore, the interruptive and noninterruptive testing entails very thorough analysis of its own internal operation and alsc 'ts interfacing with other devices.

The details of these test requirements can be found and reviewed in the external interface section of the ITM system specification. The remaining sectors under structural subsystems concentrate on describing the elements under the four categories for structural subsystems.

6.2.1.5.1 Core Avionics

Core avionics end items include mission processors, controls and displays, system mass memory, data transfer unit, data buses, fusion processors, and subsystem processors. For convenience, the above listed items are discussed as one group due to the similarity of BIT tests that are performed on each item. The similarity of BIT tests provides for similar interfaces between each item and ITM.

Each core avionics end item provides BIT test results to ITM for three categories of tests: (1) temperature, (2) voltage for power supplies, and (3) processor diagnostics. ITM, therefore, requires interfaces between itself and each core avionics end item to acquire the test results described above. The specific BIT tests associated with each core avionics end item can be found in the ITM system specification.

6.2.1.5.2 Sensors

Sensor external interfaces require unique individual test results to be provided to ITM for onboard testing. This is due to the uniqueness of each sensor's activity. Selection of the tests to be conducted and test results to be provided entailed the review of the electrical properties exhibited by each sensor, the feasibility of extracting the desired results, and an analysis of each element's historical record in meeting its reliability specifications. Only after this research was conducted could the desired test results be specified.

It was concluded that all sensors are to provide temperature, power supply voltages, and power supply ripple. All other test results are determined by the characteristics of the LRU and may be reviewed in the ITM system specification.

6.2.1.6 Nonavionic Systems

Nonavionic external interfaces are acquired from selected hydromechanical subsystems primarily for the purpose of displaying subsystem status in the cockpit (through the controls and displays) or providing fault detection and fault isolation assistance to ITM. Nonavionic systems and subsystems provide current status and consumables quantity level to ITM. This status and quantity data are provided to the mission computers, which interface with the following hydromechanical areas:

a. Engine/secondary power
b. Nonavionic electrical power
c. Hydraulics and landing gear
d. Fuel
e. Environmental control system
f. Liquid cooling system
g. Miscellaneous controls and mechanisms

6.2.2 Internal Interfaces

ITM is implemented as a system-level software function that uses selected system operational data, subsystem BIT data and system and subsystem environmental data made available to it. Exceptions to the above are two hardware devices: an MIU for ground crew use, and a data transfer unit (DTU) for offloading recorded data.

As a system, ITM was designed around two main software modules and one hardware device. The software modules are identified as the OTP and the IFTP. The hardware device is the MIU. The OTP is designed to support preflight and postflight testing while the IFTP is designed to support inflight testing. The MIU is incorporated to store postflight maintenance data for the review by the maintenance crew or pilot.

The design of the system avionics architecture and the operation of the ITM software modules required the transfer of system and fault data among the three preflight, inflight, and postflight operating modes. With each mode performing a similar but distinct function, the previous mode's results must be made available to the subsequent mode through an internal interface.

The operational makeup of the internal interface is illustrated in figure 17 and briefly described below.

6.2.2.1 Operational Procedure

At the startup of the preflight mode of flight operation, the OTP preflight mode is loaded into the mission computers and fusion processors from the system mass memory (SMM). This ITM program is selectable for loading by the pilot or ground crew. Onboard testing is initiated automatically. During the preflight operation of the OTP, system exercises are performed, reasonableness tests are conducted, and BIT data are collected by ITM to monitor for failures.

When a failure is detected, failure data are recorded within mission computer memory and on system mass memory. These failure data include such parameters as test failed, operating mode, system status, system configuration at the time of failure. ITM uses the failure data to assess the system status (is the system, functional subsystem, or structural subsystem still operational) and to support avionic reconfiguration.

Throughout the preflight mode of operation, failures are detected, failure data are recorded, and system status changes and the system configuration is modified to provide the greatest mission capability with the available avionics.

At the conclusion of the preflight mode of operation, a snapshot of the current system must be obtained in order to support the inflight phase of operation and the postflight mode where analysis of failures is conducted. With the pertinent data stored on the SMM, the IFTP is loaded into the mission computers and fusion processors, erasing the failure data that were resident in each device.

6.2.2.2 Preflight Interface Data

With the development of the ITM operational procedure came the identification of the data that are shared by and between each flight phase. In identifying the purpose of

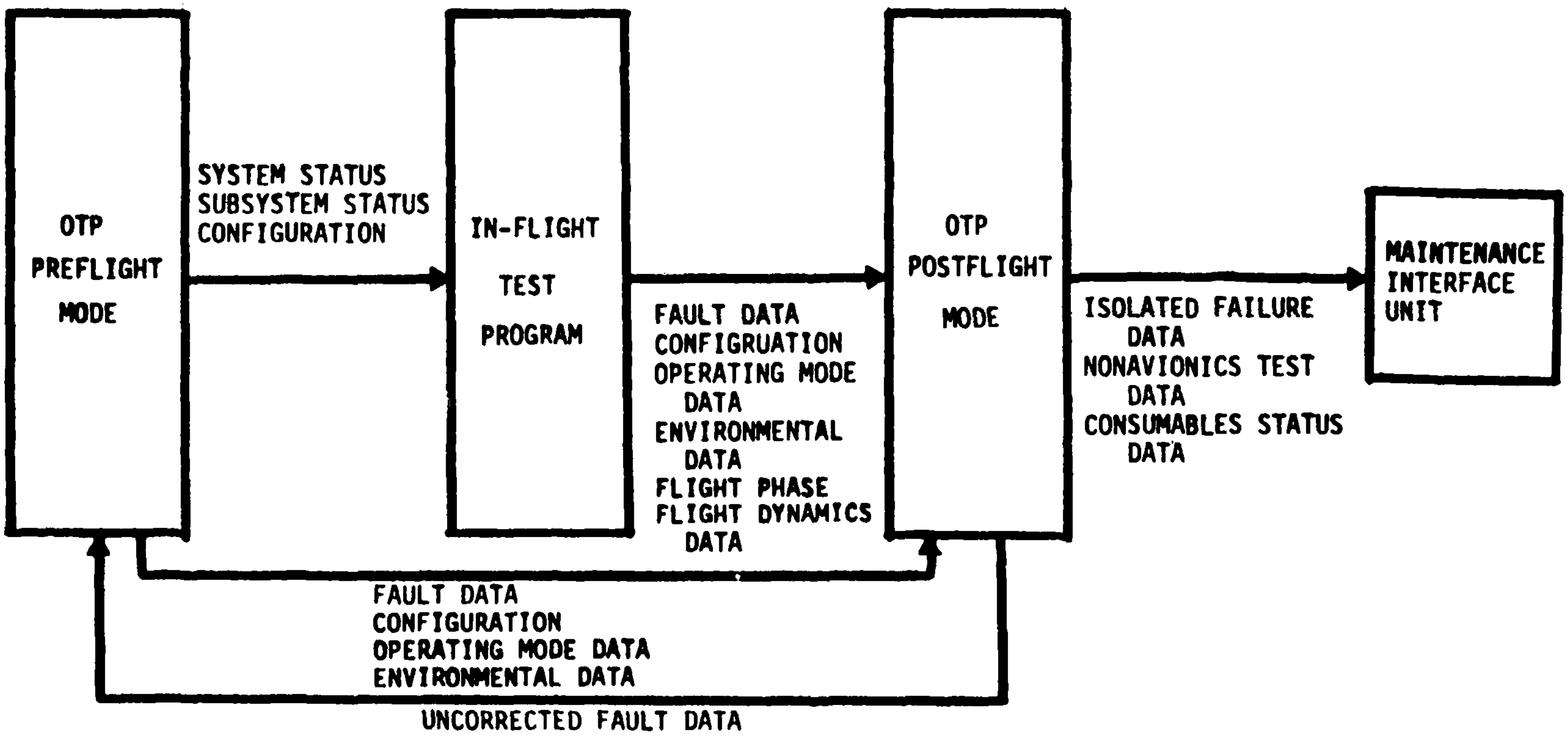


Figure 17. Major ITM Internal Interfaces

each flight phase, it became obvious that the ITM preflight mode of operation requires an interface between both the inflight phase and the postflight phase. The inflight phase requires the final results obtained by ITM in its preflight mode. This is necessary to begin flight operation with an awareness of the system limitations and of the operational status of the avionics suite.

The preflight data necessary for the inflight mode of operation are listed below:

a. System status—the final system status at completion of preflight that identifies reduction in mission capability, if any. This is required by IFTP in order to display system status after OFP initialization.

b. Subsystem status—the final subsystem status at completion of preflight that identifies any failed functional subsystems and affected operating modes. This is required by IFTP in order to display subsystem status after OFP initialization.

c. Equipment status—the final equipment status at completion of preflight that identifies any failed structural subsystems and affected operating modes. This is required by IFTP in order to display equipment status after OFP initialization.

d. Configuration—the final configuration of equipment and data buses at completion of preflight. This is required to properly initialize testing after OFP initialization.

The ITM postflight mode of operation required an extensive array of data from the ITM preflight mode. This is because postflight supports corrective maintenance by diagnosis of hard faults and by association of failure to flight phase. This diagnosis requires sufficient data to reconstruct the failure if necessary, and at the very least, enough data to identify the failure and to assist in determining common elements of failure. With preflight being performed while the aircraft is stationary and with many functions nonoperational (such as weapon release), many data that will be required from inflight are not required during preflight.

The preflight failure data to be provided to ITM postflight for failure analysis are listed below:

a. Fault data—the fault data associated with each detected failure. This is to allow isolation to the LRU.

b. Configuration—the system configuration each time it changes. This is to assist in isolation to the LRU.

c. Operating mode data—the structural subsystem operating modes at the time of any change in operating mode. This is to assist in isolation of any mode-dependent failures.

ITM preflight is also on the receiving end of data transfer. It is understood that the postflight phase of operation will not be successful in correcting all faults detected during preflight and inflight. Because of this, a turnaround preflight mode must be made aware of what equipment is not operational or is in a degraded state. This permits comparison of failures detected in subsequent preflight tests to any uncorrected failures.

6.2.2.3 Inflight Interface Data

The inflight test phase of operation provides an extensive array of data to the postflight ITM phase. It is expected that many of the detected failures and most of the intermittent failures will be encountered during the inflight phase of aircraft operation. The OTP postflight mode, therefore, demands that thorough accounting of each failure be provided to it in order to perform failure analysis and to support the development of a fault history.

During this inflight phase, environmental data, considered to play a major part in intermittent failures, must be recorded in order for postflight analysis to simulate the flight environment.

The inflight interface to postflight will require the following flight data:

a. Fault data—the fault data associated with each detected failure. This is to alow isolation to the LRU.

b. Configuration—the system configuration each time it changes. This is to assist in isolation to the LRU.

c. Operating mode data—the structural subsystem operating modes at the time of any change in operating mode. This is to assist in isolation of any mode-dependent failures.

d. Environmental data—selected environmental data when a failure is detected. This is to assist in isolation of any environmentally dependent failures.

e. Flight phase—changes in flight phase at the time of each change of flight phase. This is to determine any dependency of flight phase on failures.

f. Flight dynamics data—selected flight dynamics data when a failure is detected. This is to assist in isolation of any flight-dynamics-dependent failures.

6.2.2.4 Postflight Interface Data

As mentioned earlier, the OTP postflight mode of operation must provide to the OTP preflight mode of operation all uncorrected fault data to permit comparisons of failures detected in subsequent preflight tests to any uncorrected failures. In addition, and just as importantly, OTP postflight interfaces with the MIU to assist the ground crew in removing, replacing, and servicing activities.

In order to keep the MIU from becoming a burden for the ground crew to use, care was taken in identifying the types and quantity of data to be loaded from OTP postflight to the MIU. It is intended to give the crew a quick look at the status of the aircraft and allows them to take further action as the situation dictates.

In considering the purpose of the MIU, the aircraft information to be provided to the MIU is listed below.

a. Isolated failure data—identification of which LRU's contain failures and that need to be replaced.

b. Nonavionics test data—identification of any nonavionics items that require maintenance.

c. Consumables status data—identification of any consumables that need maintenance attention.

6.3 TESTING APPROACHES

The ITM testing approach is to supplement LRU BIT with reasonableness tests of operational data and statistical parametric testing. Preflight testing will use system exercise routines to stimulate hardware and software and allow errors to be detected by BIT or reasonableness tests. Statistical parametric testing will be used in flight to detect minor degradations in sensors.

The implementation of supplemental tests at the subsystem level is based on studies showing that additional BIT-implemented fault detection and isolation capability requires a disproportionately higher amount of dedicated hardware.(4) Subsvstem level tests, however, use existing system resources (memory, bus capacity, and throughput), which are more available due to ongoing technology advances. BIT in the ITM system will have about the same capability as systems like the F-16 and F-18, but it will have some modifications that will reduce intermittent fault indications and aid in explaining CND conditions.

6.3.1 Built-In Test

BIT at the LRU level will be modified to support ITM requirements of intermittent fault handling and fault isolation. ITM will not require the basic level of fault detection to be higher than in current fighter avionic systems. However, conditions in the LRU will be more visible at the system level so that fault indications may be analyzed and filtered before they are reported to the pilot, crew, or fault tolerance subsystem.

6.3.1.1 Role of the LRU

The LRU in the ITM system must have a BIT fault detection capability. The method of attaining this capability is left to the box designers; however, the BIT should be broken into initiated and noninterruptive tests. The latter run continually as part of background processing whenever the box is in operation and report failures in a special BIT register. Initiated (or interruptive) BIT testing is more thorough and requires the LRU to be taken offline; it is done routinely as part of preflight checkout and on pilot demand during flight.

Fault diagnostic capability in current systems is reduced because when tests within LRU's which compare a value to a set of limits (such as power supply voltage) fail, there is no indication of the amount by which the test is out of tolerance. Likewise, there is no indication of the amount by which an LRU is passing a test to limits. For the ITM system, each LRU is required to send the test data to the mission processor for those tests that are compared to a set of limits. The test results are compared to the limits stored in the mission processor. During isolation, current test data from the LRU can be compared to test data acquired in flight and stored on system mass memory.

Another benefit of testing limit data at the system level is the ability to change test limits that have been inaccurately set in the design phase. Also, as equipment ages, test limits may be changed to account for degraded operation of equipment. This allows suppression of alarms for which no maintenance action is required. A change of the test limit requires a change in the software data base rather than a hardware modification.

The BIT test in the LRU should be designed so that it is not affected by input data from other boxes. For instance, if a box is sending erroneous, noisy analog signals to other boxes in the system, the BIT in the other boxes must not indicate failure after processing the input signals. This requirement responds to the common field experience in which a single box failure causes multiple LRU's to indicate a failure and thereby reduces the ability of the system to isolate the fault.

The BIT data collection scheme for the advanced system is shown in figure 18. The MIL-STD-1553B status word contains message error, and parity error bits to signal errors in the incoming message. The terminal flag bit signals a failure of the remote terminal (or bus control unit.) Details of the specific nature of a message error as terminal flag are stored in the remote terminal BIT word. A full BIT test of the remote terminal is initiated by an initiate self-test mode code, but noninterruptive testing results are also available. The ITM core can acquire the BIT word via the transmit BIT word mode code. It should be noted that the BIT word for an imbedded remote terminal or bus control unit does not contain the status of its host processor.

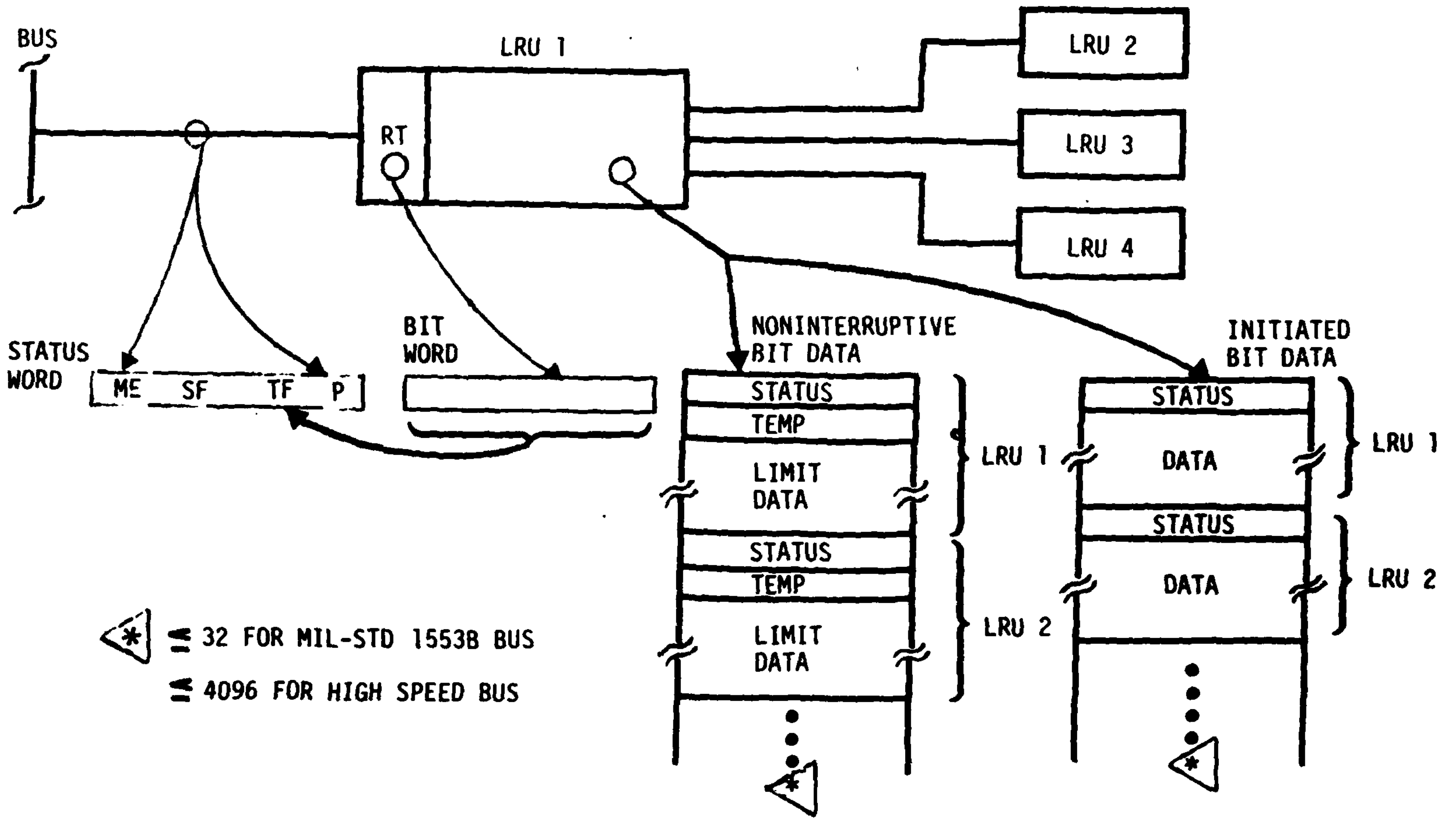


Figure 18. BIT Data Acquisition Scheme

The acquisition of noninterruptive BIT data from the host processor and any attached sensors or processors is completed by a single command and response. Address tables in the remote terminal allow for sequential transmission of status, temperature, and limit data from each of the devices in the structure.

Initiated BIT data are acquired by a separate command. These data are not normally processed in flight so they are not included in the synchronous data scheme. Interruptive testing can be initiated by sending a command to the box; the structure of the command is left to the box designers.

6.3.1.2 Role of the ITM Core

The ITM core function collects BIT data once each major frame. The master processor receives BIT data reporting the status of each LRU.

Data are collected through synchronous bus transmissions and test data that are to be evaluated against high and low limits are processed during the next frame of processing.

6.3.2 Reasonableness Tests

Reasonableness testing has found limited use in performing fault detection and fault isolation measures in currently available tactical and strategic aircraft. This is primarily the result of the attitude that BIT performs adequately in regard to ensuring acceptable test performance in addition to the fact that previous avionic architectures were structurally unique. This means that each function within the avionic subsystem utilizes unique equipment to perform the required activity. In this environment, no functional lines are crossed with avionic hardware.

Today, however, with the advances in computer technology, more powerful avionic tasks can be performed than with the hardware previously available. In addition, multiple sources of similar mission operational data have been created using this new approach to avionic design. This has then increased integration of functions and set the stage for a new category of onboard testing to fill the void created with this new class of avionic architectures.

6.3.2.1 Definition

Reasonableness tests are those onboard tests that verify the data integrity of functional subsystem mission operational parametric data. The tests were designed to detect anomalous trends in data due to failures that have not been detected by BIT. As mentioned above, reasonableness tests follow the structure imposed by the particular functional subsystem. This structure or form is derived from the subsystem functional flows illustrated in appendix B.

Because of the uniqueness of each mission flight phase, separate reasonableness tests have been identified for both preflight and inflight phases of mission operation. However, no reasonableness tests are identified for the postflight phase of operation. This is because postflight activities support analysis of test data and results obtained during preflight and inflight.

6.3.2.2 Categories of Tests

The many reasonableness tests identified in both the preflight and inflight phase of aircraft operation were constructed out of general classes of tests. These classes of tests took into account the varied results desired to assist in fault detection and isolation and the new schemes of avionic design that created, among other things, multiple sources of data that could be used for optimum and degraded mission operation.

The classes of identified reasonableness tests are described below.

a. Voting tests—these tests compare measurements of the same variable that originate from two or more different sensors.

b. Excessive deviation from average (EDA) tests—these tests compare each new piece of operational data to a running average of data values. The test is failed if the difference between the average and the new value exceeds a predefined limit for an unreasonable length of time. After each test is performed, the average is updated with the new value.

c. Limit tests—these tests detect failures by determining if any data indicate that the system is operating outside its operational envelope or environmental sensors indicate unreal conditions. The limit in these tests is predefined and fixed.

d. Rate of change (ROC) tests—these tests calculate the time derivative of a nominally continuous data stream. If the derivative exceeds a predefined limit for an unreasonable length of time, the test fails. In addition, the test will fail when the data stream does not change for a specified duration.

e. Order of command tests—these tests compare the sequence of commands received at an LRU to a predetermined sequence stored in memory. The test is failed if the order of the entries of the command is not identical to the predefined order.

f. Visual tests—these tests are actions taken by the pilot or maintenance crew to visually ensure successful completion of a reasonableness test. It will include monitoring cockpit displays and observing aircraft control surfaces. The successful completion of a test will require no further action. The failure of a test will require the pilot and maintenance crew to acknowledge to ITM that a failure has been observed.

6.3.2.3 Application

As stated in the previous sections, reasonableness testing supports functional subsystems as a class. The selection of specific reasonableness tests was undertaken through the analysis of each subsystem functional flow. There are, however, many questions that had to be proposed and answered prior to the start of test definition:

a. What parametric data must be tested?

b. What type of result is being looked for?

c. Which of the above listed categories of tests provides the desired results?

d. How does testing this parameter affect the overall fault detection and fault isolation percentages?

e. Is it physically possible to extract the necessary data to perform the test?

f. Will the proposed test overlap results obtained in other tests?

g. Does the hardware to be tested extend past proposed functional boundaries?

h. Are all critical activities in the functional subsystem tested?

i. Considering the extent of BIT, are all necessary structural and functional tasks addressed?

j. Is the identified parameter or activity testable in preflight, inflight, or both?

With a resolution arrived at in answering the above questions, reasonableness tests for each of the functional subsystems were constructed. The selected reasonableness tests can be reviewed in the ITM system specification.

It must be pointed out that the selected reasonableness tests are the result of the architecture proposed in the MAADS contract, avionic selection as derived from the Air Force and mission scenarios, and the construction of the functional flows of each subsystem identified by Boeing. With modifications to the baseline just listed, the reasonableness tests must also be reviewed and modified as necessary.

6.3.2.4 Location

The location of the identified reasonableness tests lie primarily in the mission computer. However, due to the nature of some tests, it is necessary for these tests to be performed in more local fusion or subsystem processors. Through analysis of the avionic architecture, and data transmission capabilities (i.e., high speed bus or medium speed bus) the location of each test was decided upon. The result of that analysis is shown in appendix D.

6.3.3 System Exercises

The purpose of system exercises is to ensure that the mission computers, fusion processors, controls and displays, and avionic sensors—including all avionic LRU's and data buses—perform as required when exchanging data in a fully operational environment. The previously described tests require the interaction of structural and functional subsystems to obtain the desired results. However, during preflight, most equipment is idle, with no communication or reception of pertinent data from other avionic equipment. Therefore, during preflight only the manual exchange of mission operational data is induced by the operation of system exercises.

6.3.3.1 Categories of Exercises

To ensure that all avionic activities are tested during the preflight phase of operation, categories of exercises were identified. These exercises do not address the criteria necessary to declare a test passed or failed. The success criteria and fault monitoring tests used to detect anomalous trends in data due to system failures are provided by BIT data collection and reasonableness tests. In addition, all system exercises are required to run concurrently to verify the loading capacity of the MIL-STD-1553B bus and the high-speed bus.

a. Mission processor Bus Control Interface (BCI) to BCI test—these tests transfer data between mission computers at normal bus rates to verify that addressing and formatting of data is compatible.

b. Fusion processor BCI to BCI test—these tests transfer data between fusion processors at normal bus rates to verify that addressing and formatting of data is compatible. Each fusion processor will address all other fusion processors to ensure total system compatibility.

c. Mission processor BCI to fusion processor BCI test—these tests follow functional subsystem lines of communication using actual data transferred between units. Processed data from fusion processors will be transferred to the mission computer.

d. Sensor to destination tests—these tests use either real or predefined strings of data transmitted to the sensor destination to verify source and destination transfer capability. Sensors that can transmit real data while on the ground will transmit that real data. Sensors not capable of transmitting real data while on the ground will use predefined data loaded by ITM software.

e. Mission processor to LRU tests—these tests verify the communication path between the mission processors and the LRU's associated with the avionic function of that mission processor. In addition, communication will be verified between LRU's and the backup mission computer to ensure that reconfiguration will not violate the integrity of the system.

f. Controls and displays passive and interactive tests—these tests transfer data from selected mission processors, fusion processors, LRU's, and sensors to the avionic controls and displays. Passive tests will be transparent to the pilot and crew. Interactive tests will prompt the pilot or crew for a response from incoming data and will use that response to initiate tests back to the originating source.

6.3.3.2 Selection of Exercises

The selection of system exercises is based on the functional subsystem block diagrams shown in appendix B and the support of reasonableness and BIT described in the ITM system specification. The functional flow diagrams identify the transmission paths of the avionic equipment while the reasonableness tests and BIT tests identify the parametric data that must be moved. With this data available, exercises were selected to encompass the reasonableness tests, along with the structural data identified in the selection of BIT data. The correlation of system exercises to reasonableness tests, parametric data, and data recording can be reviewed in table III of the ITM system specification.

6.3.4 Advanced Testing Techniques

A number of advanced testing techniques were considered for implementation into the ITM system. Most of the tests considered were statistical parametric tests—designed to detect minor degradations in sensors over extended periods of operation. Other tests were hardware implemented—such as signature analysis or the d-algorithm. Also, voting tests were considered as a means of testing redundant data sources. From the set of thirteen candidates, five techniques were selected for implementation in the advanced system.

6.3.4.1 Selection of Test Technique Candidates

Test techniques were analyzed to determine candidates for the advanced avionic system. The mathematical quality of each test was weighed with system design considerations to determine which techniques to develop for the advanced architecture. The test techniques considered for development were voters, least squares approximation, statistical linearization, stochastic approximation, signature analysis, d-algorithm and data compression. Additionally, several tests on the Kalman filter innovations vector were considered. These included: whiteness testing, sample mean, T^2, sample covariance, covariance signature, and pattern recognition (for classification of failure sources). Rating criteria used were those outlined in the On-Aircraft Testing Techniques Final Technical Report (8) and in the ITM Technical Proposal. The criteria were—

a. Adaptability—the ability of a testing procedure to apply to the different architecture configurations that occur because of different missions or because of system modifications.

b. Range of failures detected—the variety of failures that the technique will correctly detect or isolate. These include stuck-at faults, bridging faults, intermittent faults, soft faults, and degraded conditions.

c. Data collection cycle—the length of time a test must run to gather enough data to produce reliable results.

d. Isolation level—the level (chip, card, box, subsystem, system) at which faults are isolated by a test. Lower level isolation indicates a better test.

e. Availability of data inputs.

f. Fault detection capability.

g. Fault isolation capability.

h. False alarm immunity—the ability of the test to filter false alarms.

i. Processing time—time used to execute the test.

j. Memory requirements—the mass memory and/or processor memory required to perform the test.

k. Bus loading—additional bus use resulting from test implementation.

l. Apparent applicability—the degree of obvious utility of the test in the advanced avionic system.

m. Hardware modification—no hardware design requirements are to be levied by the test. These requirements are outside the scope of ITM.

n. Ease of development—this factor reflects limiting aspects of development costs. Some testing technique concepts are mathematically immature or require greater definition of avionic operating procedure than was available at development time.

o. Necessity—this factor is used to weigh more heavily those tests which have no substitute or those which must be run as a precursor to other tests.

Figures 19 and 20 show how these criteria were applied to each of the tests to determine which should be developed for ITM. First the quality of each test was estimated by applying criteria a through k. Next, test design considerations (criteria l through o) were factored into the figure of merit.

Those advanced testing techniques to be pursued are whiteness tests, sample mean tests, covariance signature tests, pattern recognition, and voters. Voters will be applied throughout the system, whereas the statistical techniques will focus on the navigation subsystem. Test techniques that were not developed due to lack of development time were T^2, sample covariance, least squares, and data compression.

6.3.4.2 Advanced Testing Techniques Preliminary Design

The advanced testing techniques described here are those most suitable for the advanced system architecture. Most are statistical tests (e.g., whiteness, sample mean, covariance signature tests, and pattern recognition) that apply primarily to the navigation subsystem. Voters, which are also described here, are deterministic tests that may be applied to a wide variety of system elements.

Statistical testing techniques address the problem of degraded sensor conditions that are not detected by BIT or recognizable through communications errors. They monitor mission operational data for trends that do not occur in a fully working subsystem. Once an anomaly has been identified, it must be classified and isolated•by pattern recognition procedures.

	TEST \ CRITERION	ADAPTABILITY	RANGE OF FAILURES DETECTED	A PRIORI KNOWLEDGE	ISOLATION LEVEL	AVAILABILITY OF DATA (INPUTS)	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	PROCESSING TIME	MEMORY REQUIREMENTS	BUS LOADING	TEST QUALITY	RANK
	WEIGHTING FACTOR	0.06	0.08	0.05	0.03	0.05	0.16	0.14	0.12	0.11	0.10	0.10		
INNOVATIONS VECTOR	WHITENESS	5	2	2	2	4	3	2	4	3	2	3	2.89	8
	SAMPLE MEAN	5	2	3	2	4	3	2	4	4	4	3	3.31	4
	T^2	5	2	2	2	4	4	2	4	3	3	3	2.88	9
	SAMPLE COVARIANCE	5	2	2	2	4	4	3	4	3	2	3	3.13	7
	COVARIANCE SIGNATURE	5	2	3	2	4	2	3	4	4	4	3	3.23	6
	PATTERN RECOGNITION (CLASSIFICATION OF FAILURE SOURCES)	5	3	1	3	4	4	4	4	3	2	3	3.39	2
	VOTERS	2	5	5	2	3	4	5	3	5	5	4	4.67	1
	LEAST SQUARES	3	1	2	1	3	1	1	2	2	2	3	1.80	12
	STATISTICAL LINEARIZATION	5	1	1	1	3	1	1	2	2	2	3	1.87	11
	STOCHASTIC APPROXIMATION	5	1	1	1	3	1	1	1	2	2	3	1.75	13
	SIGNATURE ANALYSIS	1	2	5	4	1	5	5	4	5	4	4	3.35	3
	d-ALGORITHM	1	2	3	5	1	5	5	4	3	2	2	3.28	5
	DATA COMPRESSION	2	2	1	3	2	3	4	2	1	2	1	2.45	10

Figure 19. Selection of Test Techniques

	DESIGN FACTORS / TESTS	TEST QUALITY	APPARENT APPLICABILITY	MODIFICATION LEVEL	EASE OF DEVELOPMENT	NECESSITY	DEVELOPMENT RATING	DEVELOPMENT RANK	DISPOSITION (PURSUE, POST-PONE, ABANDON)
INNOVATIONS VECTOR	WHITENESS	2.89	0.8	1	0.7	1.0	2.62	2	PU
	SAMPLE MEAN	3.31	0.8	1	1.0	0.8	2.12	3	PU
	T^2	2.88	0.8	1	0.5	0.7	0.81	7	PP
	SAMPLE COVARIANCE	3.13	0.8	1	0.6	0.5	0.75	8	PP
	COVARIANCE SIGNATURE	3.23	0.8	1	0.8	0.5	1.03	5	PU
	PATTERN RECOGNITION (CLASSIFICATION OF FAILURE SOURCES)	3.39	0.9	1	0.6	1.0	1.83	4	PU
	VOTERS	4.67	1.0	1	0.9	0.8	3.36	1	PU
	LEAST SQUARES	1.80	0.1	1	0.8	0.4	0.58	9	PP
	STATISTICAL LINEARIZATION	1.87	0.2	1	0.1	0.1	0.004	10	AB
	STOCHASTIC APPROXIMATION	1.75	0.2	1	0.1	0.1	0.0035	11	AB
	SIGNATURE ANALYSIS	3.35	—	0	—	—	0	12	AB
	d-ALGORITHM	3.28	—	0	—	—	0	13	AB
	DATA COMPRESSION	2.45	0.4	1	0.3	0.3	0.88	6	PP

Figure 20. Test Design Consideration

An airplane flight control loop is shown in figure 21. Airplane reactions are measured by redundant navigation sensors. The signals are filtered and integrated to determine a best estimate of position. The guidance function determines an optimum future position and the controls area produces a set of airplane control stimuli that affect the plane in the next cycle.

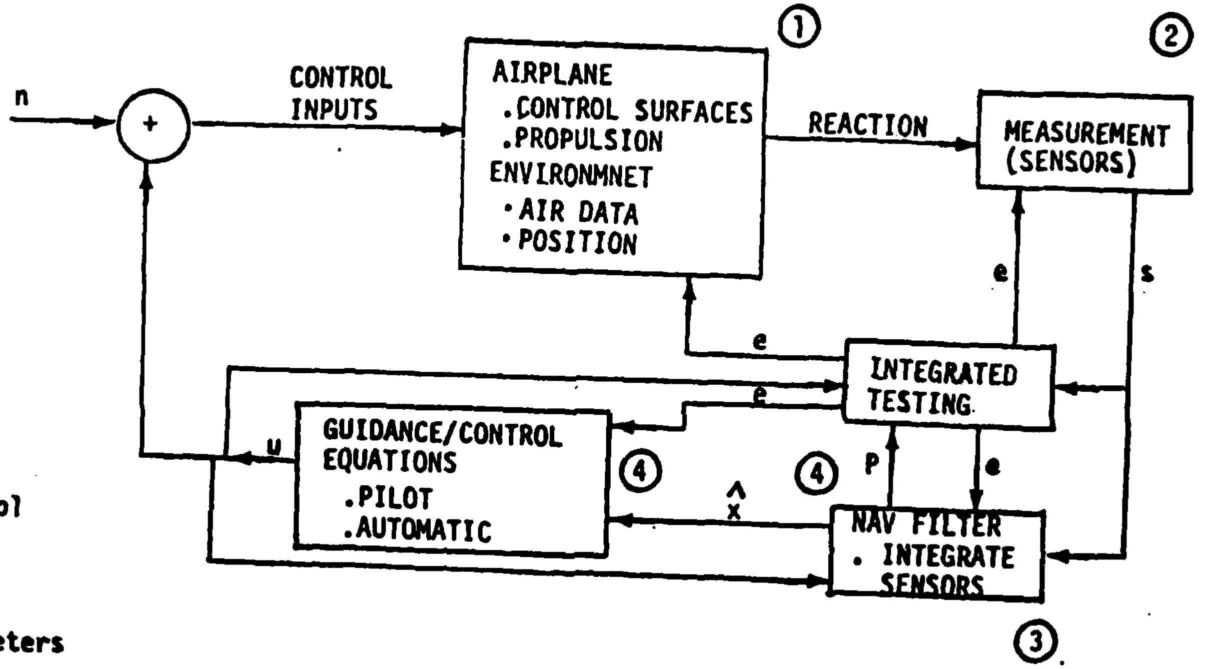
Four types of errors can cause loss of control in this system. First, a mechanical error can result in nonworking control surfaces or off-nominal propulsion. Second, measurement sensors (e.g., INS, GPS, TACAN) can fail and report low quality or no dynamics information. Third, design phase system modeling techniques may be faulty and result in incorrect estimation of the state vector. Lastly, core avionics (e.g., processors, buses) can be faulty, resulting in miscalculation or miscommunication.

Integrated testing techniques assume correct design of the filter. They must detect and isolate failures in the sensors and core avionics. Nonavionic airplane failures are not within the purview of the ITM study; however, techniques must be used that are capable of determining that they are not avionic failures.

The role of statistical testing techniques in the navigation subsystem is to monitor filter outputs to determine soft failures, degraded, and deteriorating conditions in the navigation measurement sensors. For instance, if the navigation subsystem is in integrated NAV mode, it is receiving position, velocity, attitude, and other data from a number of independent sources. The navigation Kalman filter integrates the data and determines a best estimate of the state vector (for this report, a six-element state vector is assumed). Statistical tests monitoring filter outputs identify long-term trends in the data such as biases and temporal dependencies. Many of the statistical tests concepts presented here are derived from earlier work performed by Lear Siegler (1)(8).

6.3.4.2.1 Kalman Filters

The Kalman filter used in the navigation subsystem provides a convenient statistic (or source of test data) for checking the performance of sensors, actuators, and the filter itself. As such, it is not really a testing technique but a process inherently suited to checking the sensors that provide it (the filter, or more properly, the estimator) with information from the actuators or devices—governed by the equations of motion (Kalman state equation)—and the model (including transfer functions, gains, and error covariance



s: sensor data

e: error data

$\hat{x}$: state vector estimates

u: flight control commands

n: noise

p: filter parameters

ERROR TYPES

1. Control surface and propulsion failures
2. Sensor failures
3. System modelling failures
4. Core avionic failures

Figure 21. Aircraft Flight Control Loop

matrices). The output of the filter is an estimated state vector at each sampling instant and along with the next measurement, provides the innovations vector (or vector of residuals), which is the primary test statistic. The innovations vector is just the difference between the current estimate of the state and the next measurement of the state. It has known, predictable, statistical properties; they are whiteness (independence at different time instants when the process is Gaussian), zero mean, and a known covariance matrix (computed offline).

6.3.4.2.2 Whiteness Tests

Whiteness is a property of the innovations vector sequence that indicates statistical independence of the vector at different instants in time. It is necessary to know that a process is white before further statistical testing can be performed. The failure of a whiteness test indicates that the Kalman filter does not model the system properly or that there are periodic, intermittent faults in the system.

Quantitatively, the whiteness of a process is measured by the autocorrelation function:

$$E\left\{(n_i - \bar{n}_i)(n_j - \bar{n}_j)\right\} = 0 \quad i \neq j$$

That is, at any two points in time i and j, we do not expect the deviations of the values (n) from their means ($\bar{n}$) to be related. The Fourier transform of the autocorrelation function is the power density spectrum and is constant for a white noise process.

A full whiteness test of a single element (n) of the innovations vector is

$$\bar{c}_n = \frac{1}{N} \sum_{i=k}^{N} (n_i - \bar{n})(n_{i-k} - \bar{n}) \text{ for time lag } k = 1,2. \ . \ . \ .N-1.$$

where $\bar{n} = \frac{1}{N} \sum_{i=1}^{N} n_i$ (the sample mean) and N is the number of data points collected. If the autocorrelation coefficient $\bar{c}_n$ fell within an acceptance region, the test is passed.

The number of calculations and storage space necessary to frequently perform the whiteness test for all innovation elements and for all possible time lags is very large and prohibits full implementation. The problem of finding a manageable number and duration of lags to optimize resource utilization and fault detection/false alarm parameters still remains to be analyzed.

One scheme for implementation of the whiteness test is to have a limited, iterative test running continually in the background and have a full whiteness test run when anomalies are discovered by other tests. The iterative test structure calculates the autocorrelation parameters for only two lags; it also cuts processing time by updating terms with new data rather than recalculating them each iteration.

The iterative whiteness test process is shown in table 15. The three steps in performing the test are:

a. Update the innovations vector estimate to include new data points. This step simply averages in the current vector with weight equal to previous values.

b. Calculate a new vector of autocorrelation coefficients for the given time lags (e.g., one and four cycles).

c. Test to see if the newly computed autocorrelation vector falls within the acceptance region.

The boundaries on the acceptance region are determined by the fault detection and false alarm requirements; however, they should be adjustable by the pilot or crew.

6.3.4.2.2.1 Full Whiteness Test

The full whiteness test is run when there is a high suspicion of subsystem degradation. It identifies causes of failure of sample mean tests and supplies input to pattern recognition algorithms. The test identifies nonwhite conditions for all possible lags (k) over a predetermined time interval (T).

To run the full whiteness tests, each innovations vector over the test interval must be available. The vector values are tabulated in a running window format during regular processing. When an anomaly occurs in one of the iterative tests, the full whiteness test is initiated. An estimate of the innovations vector is obtained by

TABLE 15. ITERATIVE WHITENESS TEST PROCEDURE

STORED INPUT VARIABLES	COMPUTATIONS	STORED OUTPUT VARIABLES
● STEP 1 j: index for element of vector i: current cycle index $\hat{Y}^j_{i-1}$:mean innovations vector (estimate) from last cycle Y^j_i :six element innovations vector for cycle i	UPDATE RUNNING ESTIMATE OF INNOVATIONS VECTOR: (six computations) $\hat{Y}^j_i = \frac{i-1}{i} \hat{Y}^j_{i-1} +$ $\frac{1}{i} Y^j_i$ for $j = 1,2\ldots6$	Y^j_i (replaces Y^j_{i-5} in memory) $\hat{Y}^j_i$ (replaces $\hat{Y}^j_{i-1}$ in memory)
● STEP 2 k = Lag used to test for auto-correlation $\hat{r}^j_{k(i-1)}$: estimate of auto-correlation coefficient for innovations vector element j, with lag k, from last cycle Y^j_{i-1}, Y^j_{i-4} } innovation vectors for lags of 1 and 4 cycles	OBTAIN NEW AUTO-CORRELATION COEFFICIENTS USING 2-DIFFERENT LAGS (eighteen computations) $\hat{r}^j_{ki} = \frac{i-1}{i}\left(\hat{r}^j_{k(i-1)}\right) +$ $\frac{1}{i}\left(Y^j_i\right)\left(Y^j_{i-k}\right) - \left(\hat{Y}^j_i\right)^2 +$ $\frac{i-1}{i}\left(\hat{Y}^j_{i-1}\right)^2$ $j = 1, 2\ldots6$, $k = 1,4$	$\hat{r}^j_{1i}$ (replaces $\hat{r}^j_{1(i-1)}$) $\hat{r}^j_{4i}$ (replaces $\hat{r}^j_{4(i-1)}$)
● STEP 3	THE VECTORS $\hat{r}^j_{1i}$ AND $\hat{r}^j_{4i}$ ARE COMPARED AGAINST THE HYPERELIPSOIDAL, j-DIMENSIONED ACCEPTANCE REGION. IF EITHER VECTOR FALLS OUTSIDE THE ACCEPTANCE REGION, THE TEST IS FAILED AND ANOMOLY PROCESSING WILL BEGIN	

$$\hat{Y}^j = \frac{1}{T} \sum_{i=1}^{T} Y_i^j \qquad j=1,2...6$$

where Y is an innovations vector sample, j is the element of the vector, T is the size of the test window, and i, the number of vectors in the table. Then, for each time lag (k) in the data, autocorrelation coefficients (r^i) are estimated by

$$r_k^j = \frac{1}{(T-k)} \sum_{i=k+1}^{T} (Y_i^j - \hat{Y}^j)(Y_{i-k}^j - \hat{Y}^j) \qquad j=1,2...6$$

As in the iterative test, the vector of autocorrelation coefficients for each lag is compared to the acceptance region to determine if the test is passed.

The full whiteness test consumes a relatively large amount of resources. If a six-element state vector is updated every half-second and the test interval (T) is 2 min, then the table of vectors for the test requires

(2 min) (120 updates/min) (6 elements/vector) = 1,440 floating point locations

The processing required for a single lag (k) is 1,440 iterations of the product of differences equation given above. Resource consumption rates may necessitate narrowing the window (T) or eliminating larger values for k. A possible alternative is to log the table of innovations vectors in mass memory and process it fully during postflight mode.

6.3.4.2.3 Sample Mean Tests

Sample mean tests monitor the innovations vector over time to detect deviations of the mean from zero. They are the most efficient statistical tests for detection of degraded sensor conditions. For example, if an INS gyro developed an unrecognized bias, the innovations vector elements would become biased in some manner and would no longer fall within the statistical limits of a vector with zero mean.

Two types of sample mean tests should be used. The first is a long-term test that will detect a degraded condition in the sensor suite. The test is initiated simultaneously with Kalman filter initiation. If degraded conditions are identified without isolation to a single sensor, postflight processing can cross-correlate data between modes (using an LRU data versus mode table) to isolate the faulty sensor.

Implementation of the long-term sample mean test is relatively simple. An estimate of the sample mean is updated each cycle. If the new mean vector lies outside the acceptance region, the test is failed.

The update process is similar to the first step of the whiteness test. At time i, the current six-element innovations vector Y_i is factored into the best estimate from the previous cycle and the new best estimate $\hat{Y}_i$ is derived

$$\hat{Y}_i = \frac{i-1}{i} \hat{Y}_{i-1} + \frac{1}{i} Y_i$$

The acceptance region for the mean vector is a hypersphere. The test is passed if the mean vector falls within the defined region. Using a minimum error approach to decision making, the radius of the hypersphere is determined by observing that the square of the length of the vector has a chi-squared distribution with six degrees of freedom. The significance level for the test is equal to the false alarm probability.

The second type of sample mean test is a short-term mean that is calculated from a running window of innovations vectors. The purpose of the short-term test is to quickly identify anomalous trends that indicate a deteriorating condition in a set of sensors.

The running mean is updated by recalculating the running sum and dividing by the window length; i.e.

$$RS = RS + Y_i - Y_{i-N}$$

$$\hat{Y} = \frac{RS}{N}$$

where N is the window length, and RS is the running sum.

The acceptance region can be determined in the same manner as that of the long-term sample mean test.

6.3.4.2.4 Covariance Signature Tests

Covariance signature tests are based on the principle that a change in the nature of the error covariance matrix of the Kalman filter indicates that there is a fault in the control loop, most probably an actuator or control surface failure. The FOM for the test is the signature of the estimated error covariance matrix, which is derived from the innovations sequence. The covariance signature test relies on the property of whiteness for statistical validity.

The test flow is shown in figure 22. The test is run after an anomaly has been identified by one of the iterative statistical tests. The input table of innovations vectors is the same as in the full whiteness test.

The first step is to produce the error covariance matrix. The 6 by 6 dimension of the matrix contains variances of the innovations vector elements along its main diagonal. These six elements are obtained by—

$$\hat{C}_{jj} = \frac{1}{T}\sum_{i=1}^{T} (Y_i^j - \hat{Y}^j)^2 \qquad j=1,2,\ldots 6$$

where T, Y_i^j, and $\hat{Y}$ are defined as in the whiteness test, and $\hat{C}_{jj}$ is the estimated variance of element j. The off-diagonal elements in the upper right are defined by

$$\hat{C}_{jk} = \frac{1}{T}\sum^{T} (Y_i^j - \hat{Y}^j)(Y_i^k - \hat{Y}^k)$$

where $\hat{C}_{jk}$ is the estimated covariance of elements j and k. For the lower left portion of the matrix we observe $\hat{C}_{jk} = \hat{C}_{kj}$. That is, the matrix is symmetric.

The next step is to diagonalize the matrix. This is the most time-consuming portion of the test. Row operations are used to reduce the matrix to row-echelon form (i.e., all nondiagonal elements are zero).

The signature of the matrix is obtained by summing the signs of the diagonal elements. That is, if there are four positive elements and two negative elements on the diagonal, the signature of the matrix is +2. The rank is simply the number of nonzero elements on the diagonal.

The rank and signature are compared to a nominal set, has been computed before the flight, and is based on projected airplane system dynamics. Any difference in rank or signature from nominal indicates change in the system dynamics (traceable to control surface or propulsion failures).

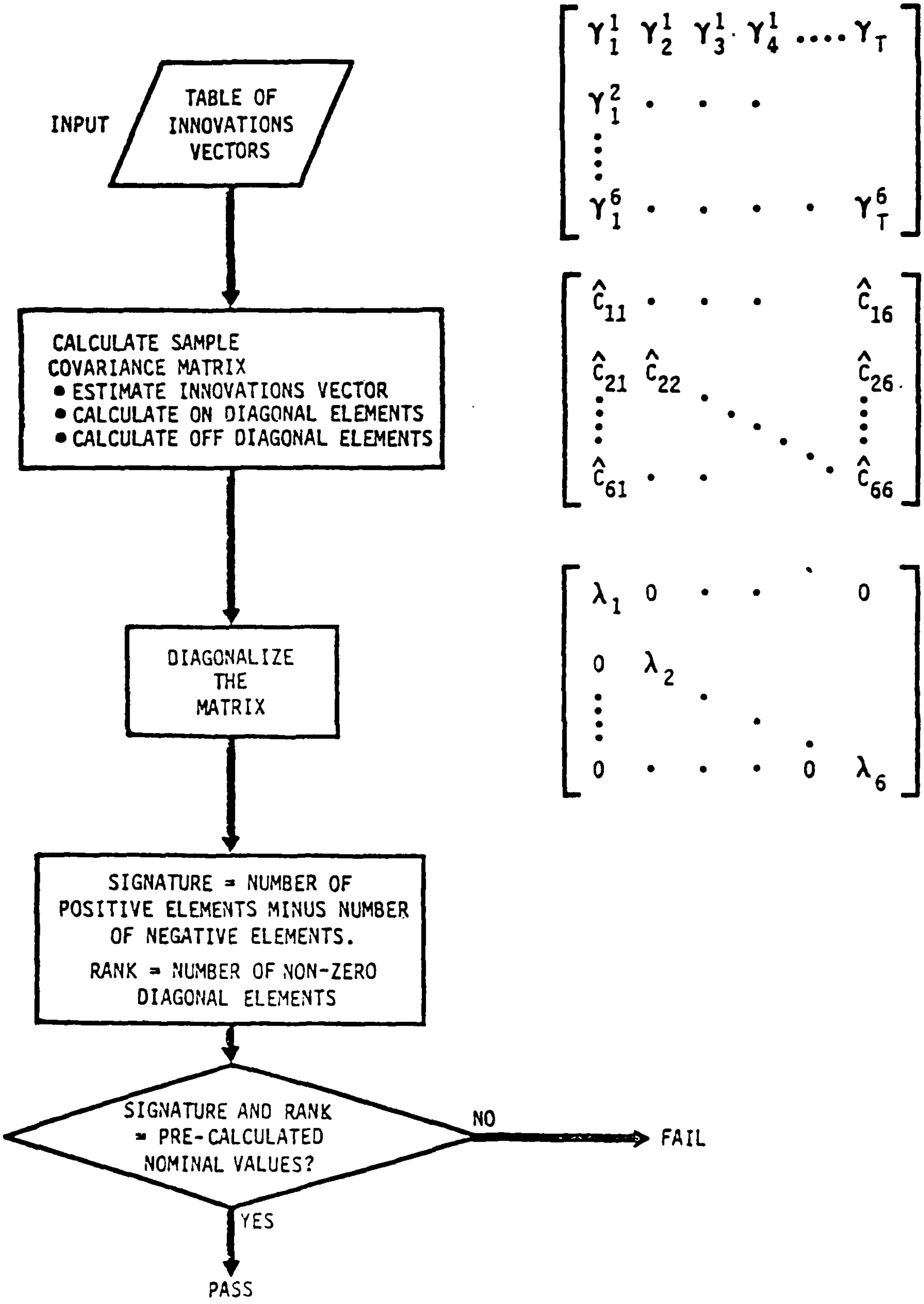

INPUT
TABLE OF INNOVATIONS VECTORS
CALCULATE SAMPLE COVARIANCE MATRIX
• ESTIMATE INNOVATIONS VECTOR
• CALCULATE ON DIAGONAL ELEMENTS
• CALCULATE OFF DIAGONAL ELEMENTS
DIAGONALIZE THE MATRIX
SIGNATURE = NUMBER OF POSITIVE ELEMENTS MINUS NUMBER OF NEGATIVE ELEMENTS.
RANK = NUMBER OF NON-ZERO DIAGONAL ELEMENTS
SIGNATURE AND RANK = PRE-CALCULATED NOMINAL VALUES?
NO
FAIL
YES
PASS
$\gamma_1^1\ \gamma_2^1\ \gamma_3^1\ \gamma_4^1 \dots \gamma_T$
γ_1^2
$\gamma_1^6\ \gamma_T^6$
$\hat{c}_{11}\ \hat{c}_{16}$
$\hat{c}_{21}\ \hat{c}_{22}\ \hat{c}_{26}$
$\hat{c}_{61}\ \hat{c}_{66}$
λ_1 0 0
0 λ_2
0 0 λ_6

Figure 22. Covariance Signature Test Scheme

6.3.4.2.5 Pattern Recognition

Pattern recognition classification functions can be used in conjunction with statistical tests to isolate faults. The problem to be solved is the lack of isolation capability afforded by the statistical tests. Pattern classification techniques use statistical test results and innovations vectors to isolate the faults. Generally, the greater the diversity of the input data, the greater the power of isolation.

A basic form of pattern recognition is illustrated in figure 23. Here, the innovations vector alone is used as the discriminant. The six-dimensional vector space is partitioned into five regions. One area represents the no-error condition and the other areas represent four types of "jump" and "step" failures. Two regions that map alterations in the state equation are—

a. Dynamic jump—an impulse in the system input that is used to model sudden shifts in bias states.

b. Dynamic step—a step function in the system input used to model an actuator failure.

Two other regions map alterations of the measurement equations. They are—

a. Sensor jump—used to model bad data points.

b. Sensor step—used to model long-term sensor failures.

Through this method, we could take an anamolous innovations vector, determine which of five areas it mapped into, and classify the source of error to be an actuator or a sensor.

Expanding the tests for further inputs allows isolation to a particular sensor or actuator. Figure 24 shows how statistical fault detection tests provide inferences of the source of failure. Given these inferences, the classification pattern recognizer will isolate to a lower level. The definition of the feature space partitioning for different failure conditions requires simulation and analysis.

a) State Equation

b) Observation Equation

v = Step siz- random variable

$\delta_{ij} = \begin{matrix} 1 & i = j \\ 0 & i \neq j \end{matrix}$

$\sigma_{ij} = \begin{matrix} 1 & i \geq j \\ 0 & i < j \end{matrix}$

H_i = hypothesis

ℓ = the point in the inno'ations sequence when step/jump occurs

$x(k)$ = n-element state vector at time k

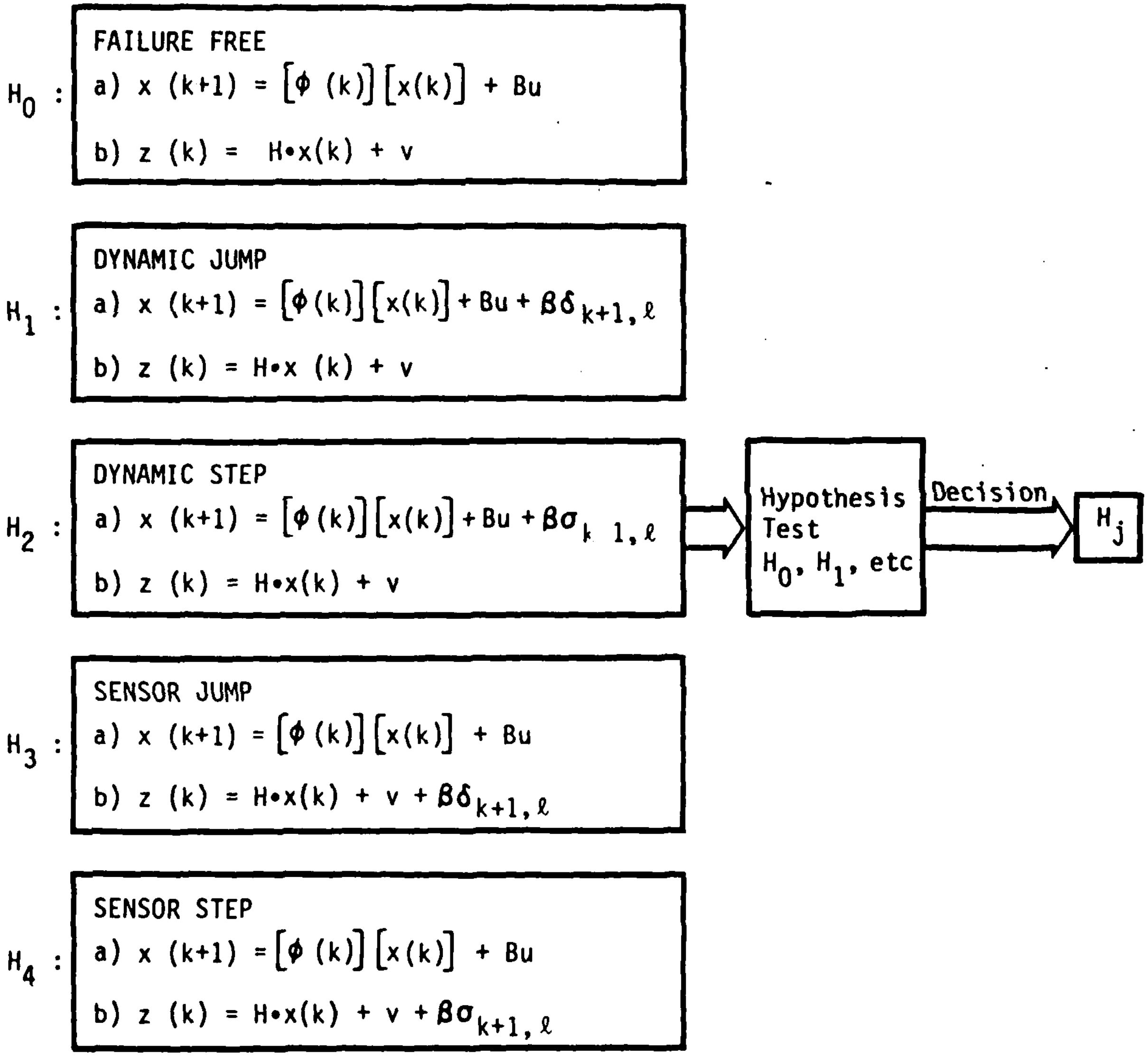
$\phi(k)$ = state transition matrix

B = n by m error propagation matrix

u = m-element error vector

H = measurement transition matrix

β = error magnitude vector



H_0 : FAILURE FREE

a) $x\ (k+1) = \left[\phi\ (k)\right]\left[x(k)\right] + Bu$

b) $z\ (k) = H \bullet x(k) + v$

H_1 : DYNAMIC JUMP

a) $x\ (k+1) = \left[\phi(k)\right]\left[x(k)\right] + Bu + \beta\delta_{k+1,\ell}$

b) $z\ (k) = H \bullet x\ (k) + v$

H_2 : DYNAMIC STEP

a) $x\ (k+1) = \left[\phi\ (k)\right]\left[x(k)\right] + Bu + \beta\sigma_{k\ 1,\ell}$

b) $z\ (k) = H \bullet x(k) + v$

H_3 : SENSOR JUMP

a) $x\ (k+1) = \left[\phi\ (k)\right]\left[x(k)\right] + Bu$

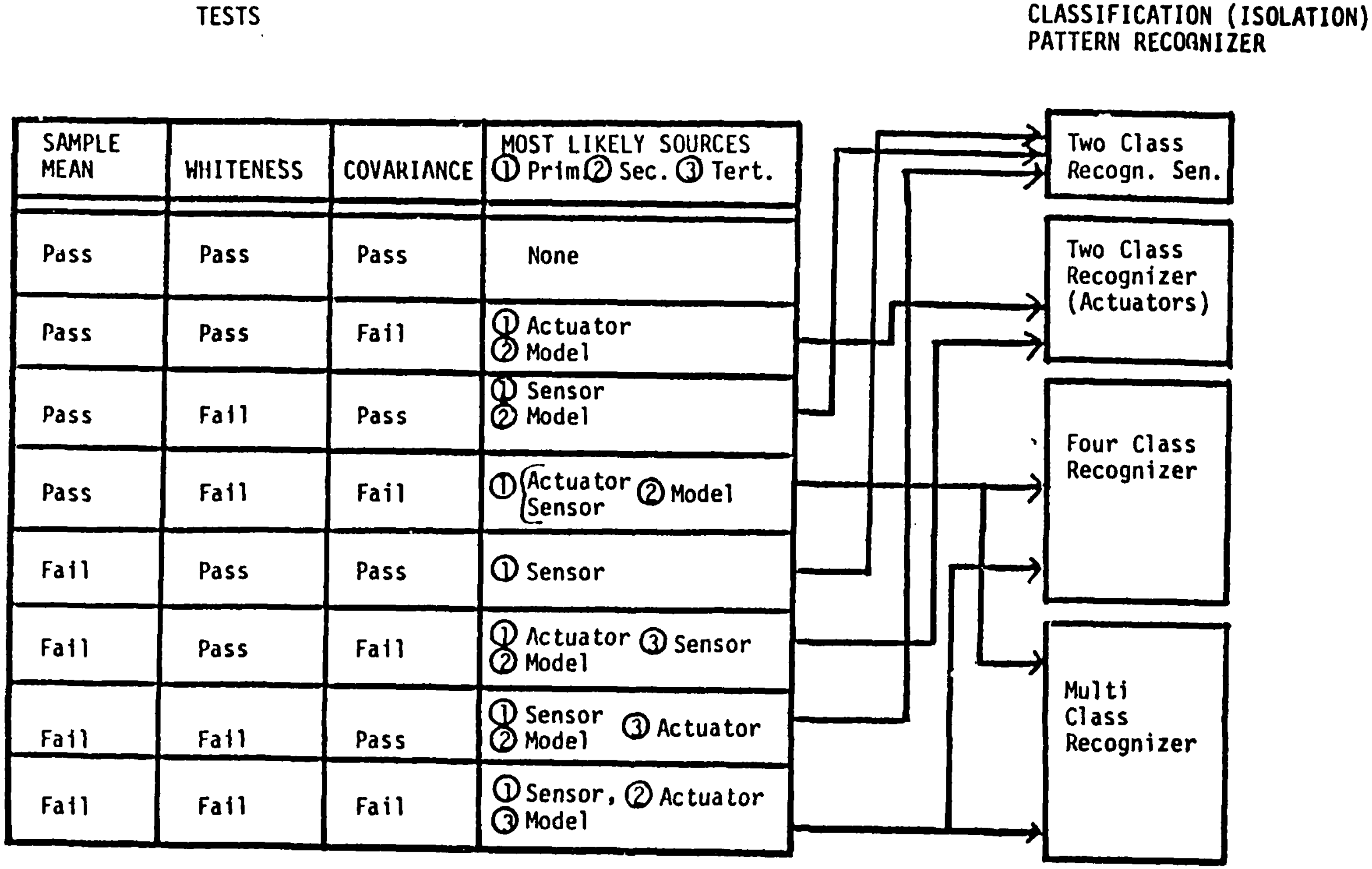
b) $z\ (k) = H \bullet x(k) + v + \beta\delta_{k+1,\ell}$

H_4 : SENSOR STEP

a) $x\ (k+1) = \left[\phi\ (k)\right]\left[x(k)\right] + Bu$

b) $z\ (k) = H \bullet x(k) + v + \beta\sigma_{k+1,\ell}$

Figure 23. Pat'ern Recognition In Classification of Kalman Filter Anomalies



SAMPLE MEAN	WHITENESS	COVARIANCE	MOST LIKELY SOURCES ① Prim ② Sec. ③ Tert.
Pass	Pass	Pass	None
Pass	Pass	Fail	① Actuator ② Model
Pass	Fail	Pass	① Sensor ② Model
Pass	Fail	Fail	① {Actuator Sensor ② Model
Fail	Pass	Pass	① Sensor
Fail	Pass	Fail	① Actuator ③ Sensor ② Model
Fail	Fail	Pass	① Sensor ③ Actuator ② Model
Fail	Fail	Fail	① Sensor, ② Actuator ③ Model

Figure 24. Source of Failure Identification

6.3.4.2.6 Voters

Voters are used to compare similar parameters obtained from two or more independent sources. Disagreements between parameters can indicate failed sensors or core elements.

Data is gathered from various points in the system and compared. If differences fall outside of tolerance levels, warning or failure flags are set.

Voting techniques are most useful for multiple, redundant sensors such as the navigation sensor suite in DAIS. The greater the number of voting entities, the higher the reliability of fault detection and isolation. At a minimum, two voters are needed for detection, three for isolation. It will be necessary to adjust tolerance levels for some voting tests (e.g., an inertial measurement unit will drift out of alignment much faster during high acceleration or angular velocity; voting tests must make looser comparisons with GPS and other high accuracy sensors during these periods).

Voters are used at many levels of processing and can be implemented in hardware or software. Some possible applications in the DAIS system are—

a. Navigation sensor suite—state vector checking for bad data points.

b. Remote terminal bus interface—identification of undetected faults in Multiplex Terminal Unit and Terminal Control Unit.

c. Processors—different system parameters could be compared between processors.

For each iteration of a voting test, nominally equivalent values are received from different sources. Values are sometimes weighted before voting takes place to compensate for differences in reliability.

6.3.4.3 Testing Techniques Evaluation

The statistical tests of the innovations vector of a Kalman filter, as developed by Lear-Siegler and discussed in the previous section, have been shown, theoretically, to be effective in detecting failures in sensors. Part of the ITM effort was directed toward evaluating these techniques by implementing them and testing their performance in a laboratory environment. This effort was discontinued because no suitable candidate could be found that was usable, within the scope of the contracted effort.

The plan for this effort was to establish a set of criteria against which various candidate Kalman filters would be evaluated, then identify and evaluate available Kalman filters. The set of criteria reflects a number of needs and concerns. They are (1) the time remaining on the contract allows only for development and evaluation of the statistical tests themselves, (2) the testbed must provide the ability to simulate many types of errors in the input sensors, (3) there is a need to work with intermediate data, (4) the testbed and filter equipment or similar equipment must be available for use, (5) the system modeled must be similar to an integrated fighter avionics system, and (6) the testbed should be tractable. In reference to these needs, the criteria in table 16 were developed.

The necessary information was gathered for a dozen Kalman filters developed by or for Boeing and one for the Air Force Flight Dynamics Laboratory. These were evaluated against the criteria in table 16. The results of the evaluation are summarized in table 17.

Although this effort could not be completed within the scope of this contract, the desirability of accomplishing this task still remains. An effort to implement and test the various statistical tests would verify their suitability for inclusion in onboard testing systems.

TABLE 16. KALMAN FILTER EVALUATION CRITERIA

IMPLEMENTATION

Item	Criteria
Number of Sensors Integrated	The statistical tests have been designed to detect and isolate faults in an "Integrated NAV" scenario. There should be a number of sensors (eg. GPS, INS, air data) blended by the filter so that the test capabilities can be fully demonstrated.
Vehicle Type	The Kalman filter's intended vehicle type should be similar to a fighter aircraft. The modeling equations of a filter vary widely in different implementations. For instance, a space system needs fewer state variables because of the simpler forces acting on it.
State Variables	The filter should be realistic; it should have enough dynamic variables and error variables to accurately etimate system parameters.
Model Complexity	The state variables should be updated such that the system is well estimated. However, it is hoped that the model will still be understandable and tractable.
Measurement and Plant Error Modeling	There are the state variables that have been incorporated in the system model that are used to track errors in the sensors and/or the control surfaces.
Computer and Language	Ideally, the computer would be a modern mainframe or mini such as the VAX 11/780, and the programs would be written in a higher order language.
Documentation	The amount and quality of documentation is a key factor in understanding a large computer program. The inputs, outputs, and process of each module should be defined.

TABLE 16. KALMAN FILTER EVALUATION CRITERIA (CONTINUED)

TEST BED

Item	Criteria
Testing Implemented	This criteria reflects the previous use of the system for purpose similar to the statistical testing task. It is important to be able to inject simulated sensor errors into the system.
Duration of Test	The time needed to run a single simulation test should not prohibit running many tests under different conditions
Error Injection	Injection of real error data into the Kalman filter model provides more certain validation of the test techniques. Theoretical error data has a tendency to help produce theorized results.
Simulation Efficiency	This is defined to be the ratio of simulated seconds to CPU *seconds* used during the simulation process. It is an indicator of how long it will take to produce data for evaluation.
Amount of Test Data	If test input data (i.e. sensor measurements) cannot be generated for any error condition of flight path, there must be a sufficient amount of this data available for use by the simulation system.
Computer and Language	Ideally, the computer would be a modern mainframe or mini such as the VAX 11/780, and the programs would be written in a higher order language.
Documentation	The amount of quality of documentation is a key factor in understanding a large computer program. The inputs, outputs, and process of each module should be defined.

TABLE 17. KALMAN FILTER CANDIDATE EVALUATIONS

Candidate	Summary
SRAM	The SRAM is a missile carried under the wing of a B-52. It was designed for use in the 1970's and therefore any models would be run on outmoded equipment and information is difficult to obtain. The Kalman filter implemented is used primarily for updating the missile before launch and not for continuous integrated navigation.
BMAC IR&D	The Kalman filter was initially developed to control the WASP missile. The program was cut before full development. It is currently being modified for use in the Advanced Cruise Missile program, but there is no baseline. Also, no simulation facility exists; testing is done on a bus, helicopter and airplane.
E-3 Tracking	The relationship between tracking hardware, faults, and software models has not been determined.
E-3 Simulation	The software was developed by a subcontractor. Access to their development models would be difficult.
E-3 Simulation	This model might have been suitable, but it has not been maintained. Costs of ressurection of the program would be fairly large.
E-3 Development	This model does not accurately depict each phase of the modeling/filtering process. It would not support broad analysis of alternatives to statistical testing. The simulations use error covariances as inputs as opposed to raw data.

TABLE 17. KALMAN FILTER CANDIDATE EVALUATIONS (CONTINUED)

Candidate	Summary
B-1A	Implemented Kalman filter model works on SRC 2000 avionics computer rather than a mainframe. This would be a very difficult system to work with. Also, the flight test data to drive the simulation has not been retained.
B-1B	Development has not been completed for the Kalman filter or the test bed. This model would probably be well-suited for statistical testing applications in the future because of the high degree of sensor integration.
BMAC R&D	Boeing Military Airplane Compnay in developing algorithms for integration of sensors in advanced fighter applications. The "sensor blending" contract has just begun and no models have been implemented in software.
IUS	The IUS Kalman filter is used only for alignment and calibration before launch. In flight, the navigation model is simplified because fewer state variables are needed in an extraterrestrial system.
ALCM	The ALCM navigation test system has most of the features that are needed for testing of statistical techniques. However, only one sensor is modeled in the system (INS). Use of this system would not provide insight into fault isolation capabilities of the tests.
ASAT	The ASAT guidance system was purchased from a subcontractor. No development systems are available in the company.

TABLE 17. KALMAN FILTER CANDIDATE EVALUATIONS (CONTINUED)

Candidate	Summary
AF Flight Dynamics Labs	A system was developed in the 1970's at the AFFDL. It was a complete test bed that included plant and model error generation, a Kalman filter, and an evaluation package. The system was used to evaluate statistical tests similar to the ones recommended in the ITM System Specification. Unfortunately, the program passed out of use and cannot now be found in card, listing, or magnetic form.

6.3.5 Fault Isolation Scheme

Once a test has failed, fault handling and isolation schemes must act to determine the source and the criticality of the fault. The ITM fault handling scheme is shown in figure 25. This scheme is used for all flight phases, except that one node (analysis by correlation) is implemented on a time available basis in flight. The end result of the fault handling scheme is to isolate the fault and tabulate its occurrence.

Following the chart, the first action after fault detection is to determine if the fault is time critical. Such a fault (e.g., a failure indicated by BIT in the Stores Launcher Processor just before weapon delivery) must be handled immediately by the appropriate subsystem. If a fault is not time critical, ITM verifies that the failure condition is still present by rerunning the failed test. The verification of the fault allows active fault isolation to occur. The active fault isolation algorithm accesses the appropriate fault diagnostic tree from system mass memory. Ideally, this results in the identification of the failed unit and fault tolerance is notified of the failure.

If the fault cannot be verified or remains present and cannot be isolated by active diagnostics, analysis by correlation begins. During flight, this consists of comparing fault occurrence times to both subsystem modes and LRU modes. Preflight and postflight correlation includes comparison of fault occurrence times to a wider range of data, including occurrence of other faults and environmental conditions.

Failure threshold tables are kept for each type of test failure. For those tests in which a single fault indication does not automatically signal the failure of a unit (e.g., bus transmission error), ITM tables store separate thresholds for number of failures, frequency of failures, and duration of failures. The three types allow for a more realistic determination of the criticality of the fault.

6.3.6 Nonavionic Testing

Selected hydromechanical systems and subsystems transfer current status and consumables quantity information to ITM. The hydromechanical areas that support nonavionic testing are listed on the following page.

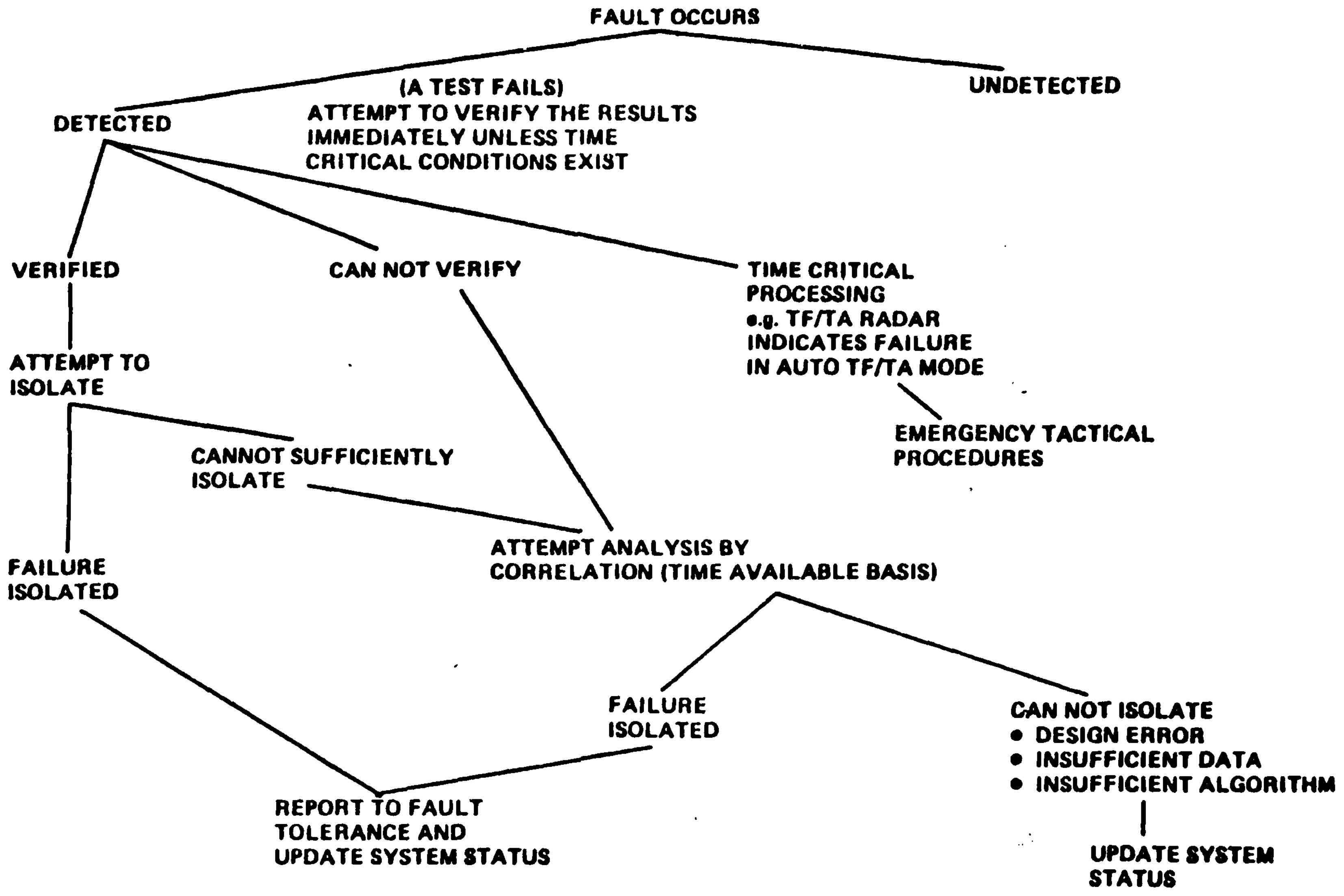


Figure 25. Fault Handling Scheme

a. Engine and secondary power.
b. Nonavionic electrical power.
c. Hydraulics and landing gear.
d. Fuel.
e. Environmental control system.
f. Liquid cooling system.
g. Miscellaneous controls and mechanisms.

The derived testing results are obtained from the BIT category of tests since the nonavionic activities are classified as structural subsystems. Review of the ITM external interfaces diagram, figure 16, illustrates the position nonavionic testing plays in total ITM testing.

The purpose of nonavionic testing is twofold: (1) to assist the pilot with information made available in the cockpit displays for cautions and advisories and (2) to assist the maintenance crew in postflight analysis. To accomplish the latter, fault data are transferred to the MIU and used to identify any nonavionic subsystems requiring maintenance. In addition, fault, status, and consumables quantity data are transferred to the DTU to be used also for offline processing.

6.3.7 Cables and Connectors

Cables and connectors are often cited as a major cause of test and maintenance problems. There are three major reasons for this. The first is the large number of cables and connections in an avionic system. Until the development of serial multiplex bus techniques, the number of cables and connections increased greater with the growing complexity of systems and subsystems. The second is the susceptibility of connections to failures. The connections can incur failures due to vibration and often as a result of maintenance (e.g., pushed-back pins). The third is that cables and connectors are overlooked in the test scheme. The tests for individual subsystems and LRU's are designed without regard for interconnections between subsystems.

Because of the use of the multiplex bus techniques the effects of the first two reasons are reduced. That is, there are fewer wires required to make interconnections and they are more secure. As for the third reason, MIL-STD-1553B and the proposed high speed bus incorporated provisions for testing the integrity of the interconnections. For these reasons, no special techniques were developed to test cables and connectors.

6.4 RELATIONSHIP OF ITM TO FAULT TOLERANCE

ITM will supply failure data to fault tolerance and will receive reconfiguration and degraded mode operations from it. Fault tolerance support will not be performed during postflight, because postflight actions are oriented towards repair rather than reconfiguration.

The following information will be supplied to fault tolerance after ITM has determined that a unit has failed: current system status, current system configuration, current configuration resources, type of fault, and fault isolation results. With these data, fault tolerance can act as a demand function, and it is able to modify the system without keeping a continual accounting of system events.

Upon completion of fault tolerance actions, ITM receives notification of changes made to the system. The modification is verified by running the test or tests that were failing prior to the modification. ITM must react to changes by testing the new units.

6.5 PERFORMANCE REQUIREMENTS

Most systems define the onboard test requirements in terms of a minimum fault detection rate, a false alarm limit, and a minimum fault isolation rate. The element of time is generally specified indirectly through specification of mean time to repair and maintenance man-hours per flight-hour requirements. Additional approaches specify maximum checkout time or limits on fault isolation time.

6.5.1 System Level Requirement

This section provides the rationale for the system performance requirements defined in the ITM system specification. In general these were developed using the F-18 performance requirements as a departure point.

6.5.1.1 Fault Detection

The fault detection rate of the system must be as close as practical to 100% without causing significant increase in the number of false alarms. The maximum rate for current systems is 98%. There is no apparent benefit to reducing this requirement, and increasing the requirement causes a disproportionate cost for the benefit. There are several categories of failures that are very difficult to detect. These include protection circuits (e.g., for noise, power protection), BIT failures that indicate good conditions, and some operating circuits, especially those at interfaces.

Typically the fault detection requirement in flight is less than for preflight and postflight. This is because during flight, the primary detection mechanism is the noninterruptive BIT of the subsystems, which provides less test coverage than the initiated BIT used in preflight and postflight. To support fault tolerance it is desirable to have the same fault detection requirements in flight as in preflight. This can be accomplished by supplementing the BIT tests by the reasonableness tests. Since the reasonableness tests that can be used in flight provide greater test coverage than the reasonableness tests used for preflight, it is feasible to provide the same fault detection requirement inflight as in preflight by using a combination of BIT and reasonableness tests.

6.5.1.2 Fault Isolation

The fault isolation requirement for the F-18 is 99% isolation to the faulty LRU. This was levied by application of the General Requirements for Maintainability of Avionics Equipment and Systems(9). This has been reduced to 98% in the ITM system specification because of the high degree of integration envisioned for the advanced avionics. This moves the ITM testability level down to the knee of the curve shown in figure 6. The remaining faults can be isolated to ambiguity groups of two or three without significant impact on maintenance time or spares.

6.5.1.3 False Alarm Rate

There is difficulty comparing false alarm requirements because of the difficulty in defining false alarms. For the purpose of specifying false alarm rates, the two categories of failures were defined, and separate rates were specified. For those caused by design defects the rate is 0%. This presumes that all false alarms of this type identified will be corrected. This can only reasonably be implemented by an evaluation period extending into the deployment and operation period of system development.

For the other false alarms, those failure indications not requiring maintenance, the requirement is 2% of the failures. This effectively reduces the CND and RTOK rates to near the 2% level. This requirement is the principal driver of the intermittent filtering and equipment temperature recording requirements.

6.5.1.4 Intermittent Faults

To reduce the false alarm rate, a requirement is needed for filtering intermittent failure occurrences to reduce alarms to the pilot. This is accomplished by establishing thresholds for number of occurrences, duration of individual occurrences, and frequency of occurrences for each failure report type. All occurrences, however, are recorded along with additional data to support postflight analysis.

The 98% isolation requirement applies to intermittent faults, but accomplishment of that level may require collection of data over more than one flight.

6.5.1.5 Preflight Checkout Time

The preflight checkout time for avionics systems is not a driver in system availability. Preflight of nonavionics systems requires a substantially longer time, and alignment of the INS (except for very coarse alignment) exceeds the avionics checkout time. The preflight checkout time was estimated to be approximately 10 min based on ITM operational sequence flows. Due to the uncertainty of some of the operating conditions, the preflight checkout time specified in the ITM system specification is 15 min. Again, this is far less than the checkout time for nonavionics and the INS alignment time.

6.5.2 Subsystem-Level Requirements

Of the above discussed requirements, only the fault detection rate needs to be allocated down to the individual LRU's and subsystems. The other requirements reflect design implementations at the system level. For allocation of the fault detection requirement to the LRU's and subsystems, three factors were considered: failure rate, criticality of the subsystem, and testability of the item.

The total system fault detection rate is the sum of individual LRU and subsystem fault detection rates weighted by their contribution to the system failure rate. As a result, the fault detection rate of low-failure-rate items can be reduced if the fault detection rate of high-failure-rate items is increased to compensate.

It is desirable to allocate a higher fault detection rate to critical subsystems and a lower rate to noncritical subsystems. This generates greater system effectiveness and availability through more effective fault tolerance. There are no critical functions in that a single failure will cause loss of mission capability. All system functions have a backup in one form or another. Criticality, then, will be determined by the criticality of the function irrespective of its susceptability to failures.

When other factors are even, it is better to allocate a higher fault detection requirement to a subsystem that is easier to test than to one that is more difficult. This generates the required fault detection rate at low cost.

An analysis was conducted to allocate the fault detection requirement to LRU's (or structural subsystems) based on failure rate, criticality, and testability. For each LRU or in some cases a structural subsystem, an assessment was made of its failure rate, criticality, and testability. These were then combined, equally weighted, into a factor for determining an appropriate fault detection rate allocation. A proposed allocation was developed based on a system fault detection rate of 98%. The top one-third of the LRU's were allocated a fault detection rate of 99% and the middle one-third retained the 98%. This permitted the bottom one-third to be reduced to 95%.

The conclusion of the exercise was that not enough variability could be obtained in the allocation process to make an allocation on these factors reasonable. This is because the system fault detection requirement is so close to 100%.

There was some merit, however, to allocating fault detection rates to the BIT capability of various subsystems based on a combination of the current test capability of similar devices and the anticipated use of reasonableness tests. The motivation was to achieve 98% fault detection in flight, but not as a requirement of the BIT. The capability is provided as a combination of BIT plus reasonableness tests and other system tests. This is an improvement over the inflight fault detection capability of 90% for the F-18. This was done for both initiated BIT and noninterruptive BIT and is summarized as follows:

Subsystem class	Initiated BIT (%)	Noninterruptive BIT (%)
Computers	98	95
Controls and displays	98	50
Sensors with substantial reasonableness test coverage	98	85
Sensors with moderate reasonableness test coverage	98	90
Sensors with minimal reasonableness test coverage	98	95
System Mass Memory	98	98
Data Transfer Unit	98	90

The lower noninterruptive BIT fault detection rates reflect the lower fault detection coverage of those tests. The 98% detection for the initiated BIT of the controls and displays includes the interactive checkout by the operator of the functions not tested by BIT. The 50% detection for noninterruptive BIT, however, recognizes that these functions are not tested. The noninterruptive fault detection rates (three levels) for the sensors accounts for relative fault coverage provided by reasonableness tests. The SMM has a higher noninterruptive fault detection rate reflecting the higher fault coverage resulting from testing of the high volume of data transfers. The DTU has a lower noninterruptive fault detection rate because the data transfers are less frequent and read or write only.

6.5.3 Software Structure

ITM software will control data collection and analysis. During preflight and postflight, ITM will be the primary function and will drive system operations. Inflight, ITM will be integrated with normal processing as an applications task. Top-level processing flow diagrams are shown in figures 26 and 27. Partitioning trade studies were conducted to determine effective algorithm implementation. Sizing and timing analysis was performed to determine the impact of ITM on inflight processing.

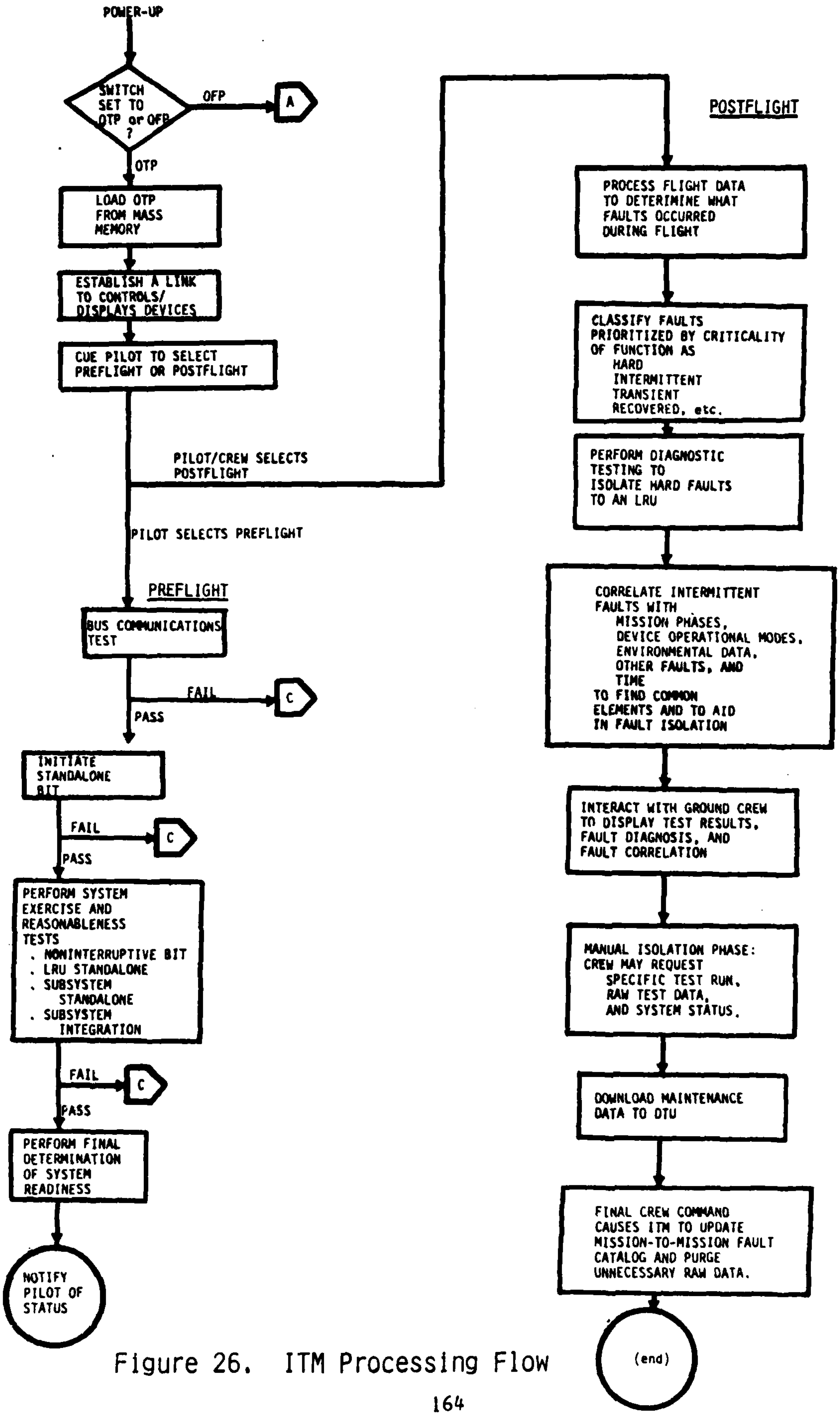

POWER-UP
SWITCH SET TO OTP or OFP ?
OFP
A
OTP
LOAD OTP FROM MASS MEMORY
ESTABLISH A LINK TO CONTROLS/ DISPLAYS DEVICES
CUE PILOT TO SELECT PREFLIGHT OR POSTFLIGHT
PILOT/CREW SELECTS POSTFLIGHT
PILOT SELECTS PREFLIGHT
PREFLIGHT
BUS COMMUNICATIONS TEST
FAIL
C
PASS
INITIATE STANDALONE BIT
FAIL
C
PASS
PERFORM SYSTEM EXERCISE AND REASONABLENESS TESTS
. NONINTERRUPTIVE BIT
. LRU STANDALONE
. SUBSYSTEM STANDALONE
. SUBSYSTEM INTEGRATION
FAIL
C
PASS
PERFORM FINAL DETERMINATION OF SYSTEM READINESS
NOTIFY PILOT OF STATUS
POSTFLIGHT
PROCESS FLIGHT DATA TO DETERMINE WHAT FAULTS OCCURRED DURING FLIGHT
CLASSIFY FAULTS PRIORITIZED BY CRITICALITY OF FUNCTION AS
HARD
INTERMITTENT
TRANSIENT
RECOVERED, etc.
PERFORM DIAGNOSTIC TESTING TO ISOLATE HARD FAULTS TO AN LRU
CORRELATE INTERMITTENT FAULTS WITH
MISSION PHASES,
DEVICE OPERATIONAL MODES,
ENVIRONMENTAL DATA,
OTHER FAULTS, AND
TIME
TO FIND COMMON ELEMENTS AND TO AID IN FAULT ISOLATION
INTERACT WITH GROUND CREW TO DISPLAY TEST RESULTS, FAULT DIAGNOSIS, AND FAULT CORRELATION
MANUAL ISOLATION PHASE: CREW MAY REQUEST
SPECIFIC TEST RUN,
RAW TEST DATA,
AND SYSTEM STATUS.
DOWNLOAD MAINTENANCE DATA TO DTU
FINAL CREW COMMAND CAUSES ITM TO UPDATE MISSION-TO-MISSION FAULT CATALOG AND PURGE UNNECESSARY RAW DATA.
(end)

Figure 26. ITM Processing Flow

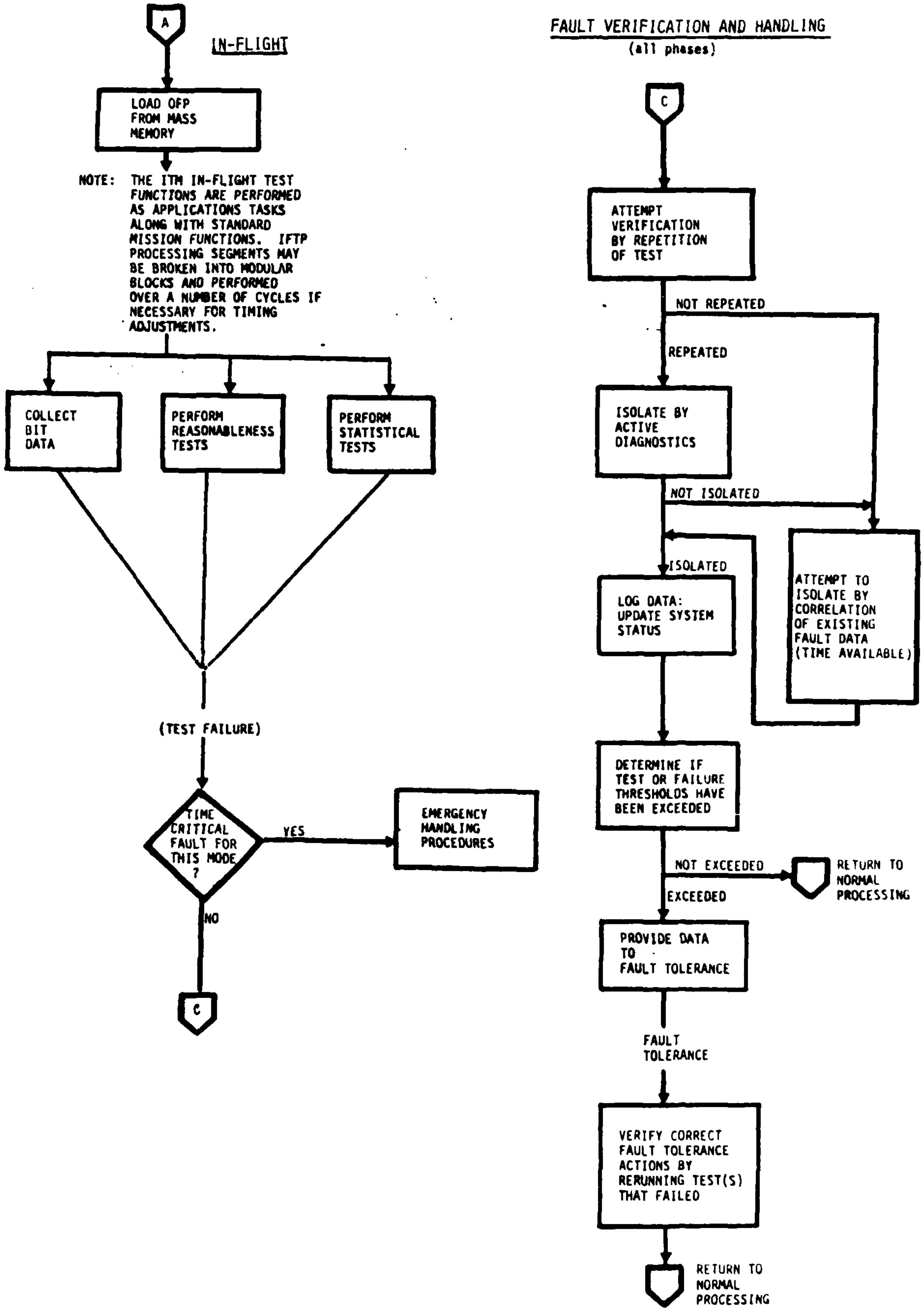

A
IN-FLIGHT
LOAD OFP FROM MASS MEMORY
NOTE: THE ITM IN-FLIGHT TEST FUNCTIONS ARE PERFORMED AS APPLICATIONS TASKS ALONG WITH STANDARD MISSION FUNCTIONS. IFTP PROCESSING SEGMENTS MAY BE BROKEN INTO MODULAR BLOCKS AND PERFORMED OVER A NUMBER OF CYCLES IF NECESSARY FOR TIMING ADJUSTMENTS.
COLLECT BIT DATA
PERFORM REASONABLENESS TESTS
PERFORM STATISTICAL TESTS
(TEST FAILURE)
TIME CRITICAL FAULT FOR THIS MODE ?
YES
EMERGENCY HANDLING PROCEDURES
NO
C
FAULT VERIFICATION AND HANDLING
(all phases)
C
ATTEMPT VERIFICATION BY REPETITION OF TEST
NOT REPEATED
REPEATED
ISOLATE BY ACTIVE DIAGNOSTICS
NOT ISOLATED
ISOLATED
LOG DATA: UPDATE SYSTEM STATUS
ATTEMPT TO ISOLATE BY CORRELATION OF EXISTING FAULT DATA (TIME AVAILABLE)
DETERMINE IF TEST OR FAILURE THRESHOLDS HAVE BEEN EXCEEDED
NOT EXCEEDED
RETURN TO NORMAL PROCESSING
EXCEEDED
PROVIDE DATA TO FAULT TOLERANCE
FAULT TOLERANCE
VERIFY CORRECT FAULT TOLERANCE ACTIONS BY RERUNNING TEST(S) THAT FAILED
RETURN TO NORMAL PROCESSING

Figure 26. ITM Processing Flow (Continued)

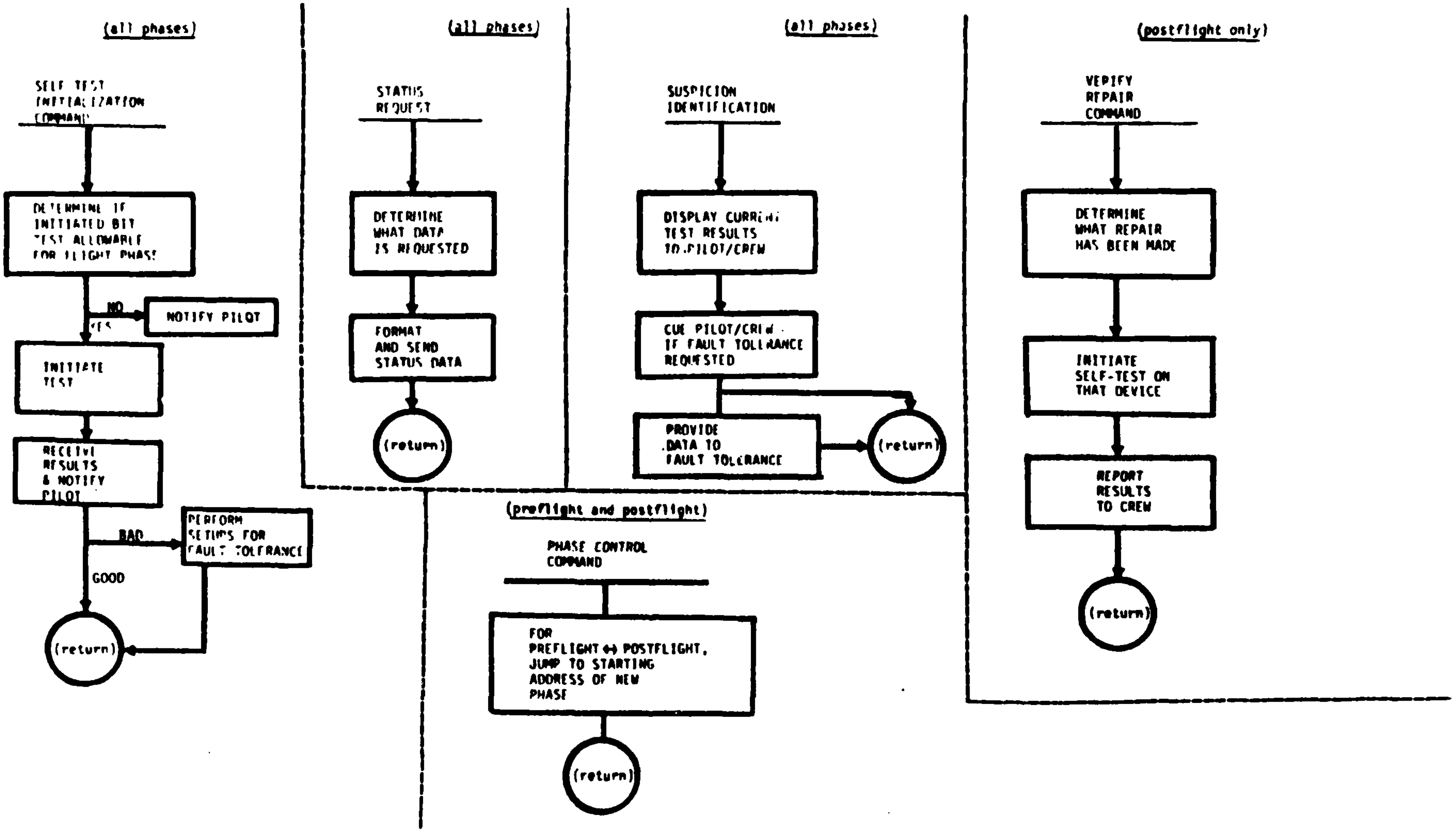

(all phases)
SELF TEST INITIALIZATION COMMAND
DETERMINE IF INITIATED BIT TEST ALLOWABLE FOR FLIGHT PHASE
NO
NOTIFY PILOT
YES
INITIATE TEST
RECEIVE RESULTS & NOTIFY PILOT
BAD
PERFORM SETUPS FOR FAULT TOLERANCE
GOOD
(return)
(all phases)
STATUS REQUEST
DETERMINE WHAT DATA IS REQUESTED
FORMAT AND SEND STATUS DATA
(return)
(all phases)
SUSPICION IDENTIFICATION
DISPLAY CURRENT TEST RESULTS TO PILOT/CREW
CUE PILOT/CREW IF FAULT TOLERANCE REQUESTED
PROVIDE DATA TO FAULT TOLERANCE
(return)
(postflight only)
VERIFY REPAIR COMMAND
DETERMINE WHAT REPAIR HAS BEEN MADE
INITIATE SELF-TEST ON THAT DEVICE
REPORT RESULTS TO CREW
(return)
(preflight and postflight)
PHASE CONTROL COMMAND
FOR PREFLIGHT ↔ POSTFLIGHT, JUMP TO STARTING ADDRESS OF NEW PHASE
(return)

Figure 27. ITM Processing Flow - Pilot/Crew Directives

6.5.3.1 Processing Flow

In the startup phase, the mission processors receive a power-up interrupt and begin power-up testing. Each processor will signal successful completion of the test by lighting a corresponding "hardware go" indicator on the Advisory Caution Panel (ACP). Upon receiving the "load" command, the master mission processor will either load the OTP or the OFP from mass memory (depending on the position of the corresponding Processor Control Panel (PCP) switch). The backup and fusion processors are also loaded.

6.5.3.1.1 Operational Test Program (Preflight)

The first action of the OTP will be to establish a communication link with the crew and determine if preflight or postflight mode is selected. If a link cannot be made or there is no response after a reasonable amount of time, the preflight program begins execution.

Preflight checkout is performed through a test hierarchy that begins at the master mission computer and progresses through system elements by testing of successive communication links and LRU's. If there is a failure reported during testing, control is passed to the fault logging and diagnostics routines. After completion of all testing and fault diagnosis and reconfiguration, the preflight program makes an assessment of which mission functions will not be available, based on its compiled tables of LRU faults and subsystem degradation. These results are reported to the pilot or crew.

6.5.3.1.2 Operational Test Program (Postflight)

After initialization of the OTP and selection of postflight mode, ITM software will begin processing the data collected during flight. The phases of postflight processing are fault classification, diagnostic testing, intermittent fault correlation, crew interaction, and data compression.

Fault classification is the first action of the postflight program. Using new fault data and system operation data that have been stored in mass memory during flight, faults are classified as hard, intermittent, transient, or recovered. Fault analysis is prioritized by these classifications. The crew may reprioritize the diagnostic sequence through the cockpit MPD.

Diagnostic testing is performed to isolate hard faults to the LRU level. There will be a diagnostic tree for each test failure, but tests may share common tree elements to save memory space.

Intermittent or transient faults that are not active in postflight are addressed in the next phase of processing. The purpose of these algorithms is to find common circumstances that existed when the faults were active. For instance, if the subsystem that controls the box was in the same mode each time the fault occurred or if the fault occurred only when temperatures were above a certain level, that would be a clue to the nature of the fault.

During the next phase of operation, ITM displays the diagnostic results to the crew and allows the crew to manually check fault data. The software downloads a list of all failed LRU's to the maintenance panel along with nonavionic data. A prompt is directed at the crew to determine whether manual isolation is desired. If so, the crew can gather data by requesting crew test data, results from correlation algorithms, or by rerunning subsystem initiated BIT. Crewmembers may continue manual isolation until they are confident of fault status. During this time the crew can direct ITM to download data sets to the DTU.

The final action taken by the postflight OTP is to consolidate test data collected during preflight and inflight phases. The mission-to-mission fault data catalog can be used as a reference for the crewmembers and for the fault isolation algorithms. For each fault, a record of the failure rate over the life of the plane allows crewmembers to determine whether a failure indicates any more than a random occurrence of the fault. A short term frequency (for the last five faults) is compared to the fault frequency since the beginning of the system's operation.

6.5.3.1.3 Inflight Test Program

Data collection and processing is performed in flight as a real-time applications task. Collection of BIT data, reasonableness tests, and other statistical tests are performed in separate time slots. Upon failure, the ITM system immediately checks for an emergency situation by comparing the test and mission phase against a prestored table. If the fault is not critical, then fault handling is set up as a background procedure.

Declaration of a failed unit leads to fault tolerance actions. All faults and fault conditions are recorded on SMM for postflight analysis.

6.5.3.1.4 Pilot and Crew Directives

The pilot crew directives shown in figure 27 are designed to allow user control over the ITM system and provide extra visibility of system status.

6.5.3.2 Partitioning Analysis

The distribution of ITM functions in the system will determine the resource utilization, ease of implementation, fault detection, isolation capability, and other factors. A partitioning analysis helps show the optimum implementation of the software algorithms. Studies were performed to determine (1) whether ITM software should be distributed or centralized, (2) whether BIT data will be collected directly from the units under test or from subsystems, (3) whether reasonableness tests will be performed by ITM software or by other subsystems, (4) whether limit testing of LRU data will be performed by ITM or the LRU, and (5) whether fault isolation will be performed by ITM at the system level or by functional or structural subsystems.

The method of analysis of partitioning alternatives was to score each against a standard set of evaluation criteria. The criteria were divided into three broad categories: (1) system design criteria reflect the need to judge how well an implementation can be maintained, modified, understood, or expanded, (2) performance criteria allow judgement of the implementation's ability to perform the ITM function (i.e., fault detection and isolation, low false alarms, and support of fault tolerance), and (3) resource criteria measure the impact of an implementation on, for example, processor memory and throughput. System design and performance criteria were given more weight than resource criteria because of the marked increase in system resources for the advanced architecture. Each major category along with its weighting was divided into several criteria, which were also weighted. The resulting partitioning evaluation matrix is shown in figure 28.

	SYSTEM DESIGN CRITERIA (.4) +							PERFORMANCE CRITERIA (.4) +				RESOURCE UTILIZATION CRITERIA (.2) = 1.0						
CRITERIA:	RETROFITABILITY	STANDARDIZATION	MODULARITY	AIRCRAFT INDEPENDENCE	RELIABILITY	MAINTAINABILITY	EXPANDABILITY	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	FAULT TOLERANCE SUPPORT	DATA COLLECTION BUS LOADING	DATA STORAGE BUS LOADING	PROCESSING LOAD	PROCESSOR MEMORY	MASS MEMORY	MISSION CRITICAL LOADING	OVERALL RATING
CRITERIA WEIGHT ⟶	.04	.04	.08	.04	.08	.08	.04	.1	.1	.1	.1	.04	.04	.04	.03	.01	.04	1.0
PARTITION ALTERNATIVE NO. 1																		
PARTITION ALTERNATIVE NO. 2																		
• • •																		

Figure 28. Partitioning Evaluation Matrix

Each of the final set of criteria is described below:

a. Retrofitability—the avoidance of hardware modification for otherwise off-the-shelf hardware (e.g., controls, displays, INS, FLIR).

b. Standardization—conformity to or tendency to produce uniform methods.

c. Modularity—having independent, integratable parts with standard interfaces.

d. Aircraft independence—how well a technique can be applied to other airplane avionic systems.

e. Reliability—the ability of the ITM system to stay operational when a system element fails.

f. Maintainability—ease with which a system can be fixed or altered.

g. Expandability—the ability to add or strengthen functions.

h. Fault detection capability—the speed and reliability of fault detection.

i. Fault isolation capability—the speed and reliability of correct fault isolation.

j. False alarm immunity—the ability of the system to avoid indicating errors when none exist.

k. Fault tolerance support—the speed with which fault tolerance can be notified of the correct failure area.

l. Bus loading (dynamic data collection)—the bus traffic dedicated to collection of test data.

m. Bus loading (data storage and retrieval)—data sent to and from the mass memory for auxiliary storage.

n. Processing load—throughput required for an implementation.

o. Memory utilization (processors)—data and instruction storage in the master and fusion processors.

p. Memory utilization (mass memory)—data storage in the system mass memory.

q. Mission critical loading—load of an alternative during critial flight phases and maneuvers.

The results and rationale for the partitioning studies are given below. A scale of zero to five, where five is most favorable, is used to score alternatives against the criteria. It should be noted that the relative values of the scores, rather than the absolute values, determine which criteria are to be selected. A blank in the scoring table indicates there is no clear advantage for either alternative.

6.5.3.2.1 ITM Software

ITM software will be in the master processor when possible. (Scoring is shown in figure 29.)

Alternatives studied:

a. Centralized when possible.
b. Centralized except for advanced statistical testing techniques.
c. Decentralized with absolute minimum in master.

Ground rules:

a. For option 1, the master processor performs BIT data collection and testing for all devices. All interfaces are part of the ITM core in the master processor.

b. For option 2, it is assumed that the advanced statistical testing techniques will reside in the ICNIA data processor.

c. For option 3, it is assumed that data collection and testing tasks are partitioned to the fusion processor that is best able to receive data from the unit under test. Smart interfaces are required in the area of the interfaced unit.

	SYSTEM DESIGN CRITERIA (.4)							+ PERFORMANCE CRITERIA (.4)				+ RESOURCE UTILIZATION CRITERIA (.2) = 1.0						
CRITERIA:	RETROFITABILITY	STANDARDIZATION	MODULARITY	AIRCRAFT INDEPENDENCE	RELIABILITY	MAINTAINABILITY	EXPANDABILITY	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	FAULT TOLERANCE SUPPORT	DATA COLLECTION BUS LOADING	DATA STORAGE BUS LOADING	PROCESSING LOAD	PROCESSOR MEMORY	MASS MEMORY	MISSION CRITICAL LOADING	OVERALL RATING
CRITERIA WEIGHT→	.04	.04	.08	.04	.08	.08	.04	.1	.1	.1	.1	.04	.04	.04	.03	.01	.04	1.0
1. ALL IN MASTER	5		1	4	4	4	3		3		5	2		1	1			2.13
2. ALL IN MASTER EXCEPT ADVANCED TESTING TECHNIQUES	5		2	3	3	3	3		3		4	2		2	3			2.03
3. ABSOLUTE MINIMUM IN MASTER	4		4	2	1	1	4		4		2	4		5	5			1.99

Figure 29. Partitioning Evaluation Matrix - Software

Scoring justification:

a. Retrofitability—Because neither ICNIA nor the master processor are off-the shelf, neither has impact on retrofitability. Due to intelligence addition to fusion processors, (1) and (2) have no difference, (3) is seen to have some impact.

b. Standardization—no impact; ITM will set the standard in this area.

c. Modularity—a system/subsystem/CI is more modular to the greater degree that it tests itself and reports to top-level ITM via a standard interface. It is less modular if it must interface a unique set of raw data that ITM must be modified to handle.

d. Aircraft independence—most aircraft avionic systems have a master computer. The similarity between the advanced architecture and other architectures decreased as more boxes are considered.

e. Reliability—the reliability of the ITM software system is based on the number of boxes used to implement the software.

f. Maintainability—again the centralized schemes are better because an update requires fewer fixes and there is less integration of fixes.

g. Expandability—generally, software will be more expandable if it is distributed over more units, thereby decreasing its impact on any one unit.

h. Fault isolation—the distributed scheme is slightly better because of its more direct access to LRU's. Also, the distributed scheme implies status reporting from a number of elements along a data processing path, thereby aiding isolation.

i. Fault tolerance support—the centralized scheme provides better support because the local availability of data provides for a faster assembly of interface data.

j. Bus loading-dynamic data collection—decentralization implies more bus loading because, in many cases, data would travel from the sensor to a fusion processor, which would check the data and then send status data back out over the bus to the master processor—creating superfluous bus traffic. Centralization would usually require only one bus transmission, from the unit under test to the master processor.

k. Processing load and memory—the distributed processing scheme will provide for much lower processing load and memory requirements on the central computer.

6.5.3.2.2 Acquisition of BIT Data

ITM software will acquire BIT data directly from the unit under test. (Scoring is shown in figure 30.)

Alternatives studied:

a. Acquire BIT data through non-ITM applications tasks.

b. Acquire BIT data directly from the unit under test.

Ground rules:

a. For option 1, the non-ITM applications task (e.g., target acquisition) collects BIT data from its sensors (e.g., laser designator, attack FLIR) along with operational data. It then either passes the data directly to the master processor or tests the data and passes the results to the master.

b. Option 2 has the sensor or bus interface unit—in a separate transmission—pass BIT data directly to the master processor.

Scoring justification:

a. Standardization—option 2 is more adaptable to standards because a simple transmission from a sensor can be defined by the type of sensor being inspected. Option 1 requires an additional standard for some undefined intermediate module.

b. Reliability—the direct method (option 2) is more reliable, because it requires fewer hardware boxes to implement.

CRITERIA:	SYSTEM DESIGN CRITERIA (.4) +							PERFORMANCE CRITERIA (.4) +				RESOURCE UTILIZATION CRITERIA (.2) = 1.0						
	RETROFITABILITY	STANDARDIZATION	MODULARITY	AIRCRAFT INDEPENDENCE	RELIABILITY	MAINTAINABILITY	EXPANDABILITY	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	FAULT TOLERANCE SUPPORT	DATA COLLECTION BUS LOADING	DATA STORAGE BUS LOADING	PROCESSING LOAD	PROCESSOR MEMORY	MASS MEMORY	MISSION CRITICAL LOADING	OVERALL RATING
CRITERIA WEIGHT →	.04	.04	.08	.04	.08	.08	.04	.1	.1	.1	.1	.04	.04	.04	.03	.01	.04	1.0
1. THROUGH NON-ITM APPLICATIONS TASKS		3	1		2	2						2		3	4		1	.92
2. DIRECT FROM UNIT UNDER TEST		5	4		4	5						4		4	2		4	1.82

Figure 30. Partitioning Evaluation Matrix - Acquistion Of BIT Data

c. Maintainability—the direct system is more maintainable because corrections of problems require less integration of updates in the system.

d. Bus loading-dynamic data collection—the transferral of data through an intermediate point (as in option 1) requires more bus use than the direct transfer to the master processor.

e. Processing load—option 2 will require less processing systemwide without significantly increasing the load on the master processor.

f. Processor memory—the master processor will require additional data storage area for option 2 to hold test results and intermediate data for most system elements.

g. Mission critical loading—the direct application scheme presents far less burden during mission critical phases because a lower bus traffic and ease of partial shutdown of ITM processing.

6.5.3.2.3 Reasonabless Tests

Reasonableness tests will be performed by ITM software located in the processor that has best direct contact with the equipment being tested. (Scoring chart is shown in figure 31.)

Alternatives studied:

a. Perform reasonableness tests within subsystem applications tasks and report results to ITM core.

b. Perform reasonableness tests in ITM applications tasks that will be placed in the same fusion processor as the subsystem that operates on the data.

Ground rules:

It is assumed that ITM software for option 2 will be located in the same processor as the corresponding applications task in option 1.

	SYSTEM DESIGN CRITERIA (.4)	+						PERFORMANCE CRITERIA (.4)	+			RESOURCE UTILIZATION CRITERIA (.2) = 1.0						
CRITERIA:	RETROFITABILITY	STANDARDIZATION	MODULARITY	AIRCRAFT INDEPENDENCE	RELIABILITY	MAINTAINABILITY	EXPANDABILITY	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	FAULT TOLERANCE SUPPORT	DATA COLLECTION BUS LOADING	DATA STORAGE BUS LOADING	PROCESSING LOAD	PROCESSOR MEMORY	MASS MEMORY	MISSION CRITICAL LOADING	OVERALL RATING
CRITERIA WEIGHT →	.04	.04	.08	.04	.08	.08	.04	.1	.1	.1	.1	.04	.04	.04	.03	.01	.04	1.0
1. WITHIN APPLICATIONS TASKS		4	3			4	4								4			1
2. WITHIN ITM SOFTWARE		3	4			5	5								3			1.13

Figure 31. Partitioning Evaluation Matrix - Reasonablesness Tests

Scoring justification:

a. Standardization—reasonableness tests that are part of standard functional software (1) are better than tagged-on modules not part of standard functional software.

b. Modularity—separation of operational function and test function allows easier movement of either to other processors or within processor memory.

c. Maintainability—the separate approach (2) is more maintainable because the ITM programmer needs less knowledge of the software for the operational system begin tested.

d. Expandability—independent expansion of either ITM or the operational subsystem under test is simpler if the functions were separated.

e. Processor memory—the integrated approach (1) uses less processor memory because of the overhead involved in program segments, which ensure that the routines are modular and independent.

6.5.3.2.4 LRU Level Limit Testing

LRU-level limit testing will be performed by ITM software in the master processor. Limit test values will be uploaded to ITM software in the master processor for testing. (Scoring chart is shown in figure 32.)

Alternatives studied:

a. Variable test limits will be downlinked to the LRU for each limit-tested element (e.g., voltage, temperature). The LRU will perform the test and send a pass-fail indication to ITM applications tasks.

b. Each LRU will upload data for limit testing to ITM applications tasks that will perform the tests.

	SYSTEM DESIGN CRITERIA (.4) +							PERFORMANCE CRITERIA (.4) +				RESOURCE UTILIZATION CRITERIA (.2) = 1.0						
CRITERIA:	RETROFITABILITY	STANDARDIZATION	MODULARITY	AIRCRAFT INDEPENDENCE	RELIABILITY	MAINTAINABILITY	EXPANDABILITY	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	FAULT TOLERANCE SUPPORT	DATA COLLECTION BUS LOADING	DATA STORAGE BUS LOADING	PROCESSING LOAD	PROCESSOR MEMORY	MASS MEMORY	MISSION CRITICAL LOADING	OVERALL RATING
CRITERIA WEIGHT →	.04	.04	.08	.04	.08	.08	.04	.1	.1	.1	.1	.04	.04	.04	.03	.01	.04	1.0
1. WITHIN LRU	1		3		1	1	2	2	2	1	2	4	3	4	4		5	1.98
2. WITHIN ITM SOFTWARE	3		2		3	3	5	4	3	5	3	1	1	1	2		2	2.72

Figure 32. Partitioning Evaluation Matrix - BIT Test To Limits

Scoring justification:

a. Retrofitability—Both options will require some retrofitting for off-the-shelf LRU's. However, option 1 requires more due to additional processing for the limits test. Option 2 requires only that a buffer containing the test values be made available to the bus interface unit.

b. Modularity—option 1 provides better modularity because it has fewer data interfaces.

c. Reliability—option 2 is more reliable because the decreased role of LRU processing allows for fewer faults at the LRU level.

d. Maintainability—option 1 is less maintainable because of its hardware-biased implementation.

e. Expandability—option 2 meets this criterion better because limit testing techniques could be expanded by a software change onlv rather than a probable hardware-software change for option 1.

f. Performance criteria—ITM applications testing of LRU limit data (2) has marked performance advantages. All performance areas benefit from ITM visibility of how badly a test failed or how easily it passed.

g. Data collection-dynamic data bus loading—option 1 is better because a simple go/no-go status is sent over the bus for each test rather than sets or raw data (option 2).

h. Bus loading-data storage and retrieval—coincident with the performance gains for option 2 are additional data storage and acquisition for utilization of extra data.

i. Processing load—the ITM core processor has a much larger load for option 2 because of the centralized limit testing. It requires the time to perform three limit tests per TRU per second.

j. Processor memory—high and low limits are actively stored in processor memory for option 2 along with detailed test results.

k. Mission critical loading—high resource utilization provides a greater burden during mission critical periods for option 2. There is effort involved in suppressing synchronous transfer of the limit data.

6.5.3.2.5 Fault Isolation Decisions

Fault isolation decisions will be made by ITM software through use of data from functional subsystems, structural subsystems, and LRU's. (Scoring chart is shown in figure 33.)

Alternatives studied:

a. Determine faulty LRU by structural isolation of faults.
b. Determine faulty LRU by isolation of faults within functional subsystems.
c. Isolate faults within ITM core software using both structural and functional data.

The tradeoff being made is to determine whether the extra benefits gained from integrating functional and structural data in fault isolation warrant the extra resource costs incurred in implementation.

Scoring justification:

a. Retrofitability—option 1 requires modification and expansion of BIT acquisition for remote terminals and other bus interface structures. Options 2 and 3 require no hardware modifications to implement.

b. Standardization—option 3 is rated higher because the integrated approach does not imply standards for unique functional and structural subsystems.

c. Aircraft independence—avionic structures are not common between airplane types; functions are somewhat more independent. The approach of option 3 is most independent because of the concept approach of complementary function-structure integration.

	SYSTEM DESIGN CRITERIA (.4)							+ PERFORMANCE CRITERIA (.4)				+ RESOURCE UTILIZATION CRITERIA (.2) = 1.0						
CRITERIA:	RETROFITABILITY	STANDARDIZATION	MODULARITY	AIRCRAFT INDEPENDENCE	RELIABILITY	MAINTAINABILITY	EXPANDABILITY	FAULT DETECTION	FAULT ISOLATION	FALSE ALARM IMMUNITY	FAULT TOLERANCE SUPPORT	DATA COLLECTION BUS LOADING	DATA STORAGE BUS LOADING	PROCESSING LOAD	PROCESSOR MEMORY	MASS MEMORY	MISSION CRITICAL LOADING	OVERALL RATING
CRITERIA WEIGHT →	.04	.04	.08	.04	.08	.08	.04	.1	.1	.1	.1	.04	.04	.04	.03	.01	.04	1.0
WITHIN STRUCTURAL SUBSYSTEM	1	2		1		1	1		2	2	3	5		5	5		5	1.73
WITHIN FUNCTIONAL SUBSYSTEM	4	2		2		3	3		2	3	4	3		4	3		3	2.07
WITHIN ITM CORE SOFTWARE	4	4		3		2	4		5	5	5	3		1	1		1	2.57

Figure 33. Partitioning Evaluation Matrix - Isolation To LRU

d. Maintainability—functional isolation is most maintainable because of the ease of software fixes relative to hardware fixes. Both option 1 and option 3, which has structural orientation, are less maintainable.

e. Expandability—option 3 has the greatest realm of expansion—both within functional and structural areas. Functional expansion is simpler than structural expansion.

f. Fault isolation—the integrated approach of option 3 combines information available to the other two approaches to provide more powerful and reliable fault isolation.

g. False alarm immunity—option 3 has the greatest redundancy of data and can best filter false alarms. Option 2 provides advantages over option 1 because multiple functions can have access to an LRU while an LRU is usually part of only one structure.

h. Fault tolerance support—efficiency of support is best for isolation at the highest level (i.e., the level of the interface with fault tolerance).

i. Bus loading-dynamic data collection—structural isolation utilizes bus transmissions only to initiate tests and report results. Functional and integrated isolation utilize the bus for communications between system elements.

j. Processing load—the integrated approach requires much more data correlation at the master processor level.

k. Processor memory—the algorithims necessary for the functional approaches 2 and 3 use additional memory for correlation of functional subsystems. Option 3 also correlates this with structural data.

l. Mission critical loading—fault isolation is needed more frequently during times of avionic stress. For each additional fault in that period isolation schemes absorb some resources.

6.5.3.3 Sizing and Timing

Sizing and timing estimations were performed to determine the impact of ITM on inflight processing. Using the results of the partitioning analysis and the list of processing modules necessary to perform tasks, estimates were determined for processor memory, processor throughput, bus loading, and SMM storage. A liberal amount of resources was assigned for each task, and a 20% compiler inefficiency was added for processing functions.

For each module, sizing and timing were computed on a basis of demand per fault indication and in normal (no faults present) conditions. A fault indication rate of one per minute and a 5-hr mission were assumed to find average demand levels. Table 18 shows the estimation scheme cross-reference for the inflight modules.

A summary of the sizing and timing estimates is shown in table 19. Demand throughput and bus loading have been converted from "per fault" units to "per second" units so that they can be added to the estimates for a fault-free environment. Similarly, demand SMM storage has been estimated for the entire mission by assuming one fault per minute over a 5-hr mission.

Final estimates are made by adding a 20% compiler inefficiency factor to the program instruction size, throughput, and bus loading estimates. That is:

$$\text{Final estimate} = \left(\begin{array}{l}\text{Fault-free}\\\text{loading}\end{array} + \begin{array}{l}\text{Demand}\\\text{loading}\end{array}\right)\left(1 + \begin{array}{l}\text{Compiler}\\\text{inefficiency}\\\text{factor}\end{array}\right)$$

The impact on the advanced architecture (having a MIL-STD-1553B bus, MIL-STD-1750A processor with 256K words memory operating at 1 MIPS, and a 400-MB SMM) is shown in table 20.

A detailed accounting of sizing and timing estimates for each module is in appendix C.

Table 18. Sizing and Timing Measurement Reference

	FAULT FREE ENVIRONMENT, NO PILOT INTERACTION				DEMAND PER FAULT OR PILOT DIRECTIVE		
IMPACT MODULE	PROGRAM STORAGE	THROUGHPUT	BUS LOADING	MASS MEMORY STORAGE	THROUGHPUT	BUS LOADING	MASS MEMORY STORAGE
COMPONENT CONTROL	X	X					
BIT DATA COLLECTION	X	X	X				
REASONABLENESS TESTS	X	X	X				
STATISTICAL TESTS	X	X	X	X	X	X	
INITIATED SELFTEST	X			X	X	X	X
SUSPICION IDENT.	X			X	X	X	
STATUS REQUEST	X				X	X	
SYSTEM STATUS MAINT.	X			X	X	X	
FAULT DATA RECORDING	X				X	X	X
MODE CHANGE RECORDING	X	X	X	X			
FAULT VERIFICATION	X				X	X	
ACTIVE FAULT ISOLATION	X				X	X	
FAULT CORRELATION	X			X	X	X	
FAILURE THRESHOLD TEST	X			X	X	X	
EMERGENCY HANDLING	X				X		X
FAULT STATUS	X				X	X	X
RECONFIGURATION VERIF.	X				X	X	X

Table 19. Resource Requirement Summary

FUNCTION	PROGRAM SIZE		THROUGHPUT (OPS)	BUS LOADING (WPS)	SYSTEM MASS MEM. STORAGE	DEMAND THROUGHPUT (OPERATIONS)	DEMAND BUS LOAD. (WORDS)	DEMAND SYSTEM MASS MEM. STORAGE
	DATA	INSTRUCTION						
MONITOR FOR FAULTS								
• BIT DTA COLLECTION	900	150	750	250	0	0	0	0
• REASONABLENESS TEST	3700	500	400	50	0	0	0	0
• STATISTICAL TESTS	680	2000	1000	100	5200	5000	1500	0
PILOT/CREW DIRECTIVES								
• SELF TEST INIT.	100	160	D	D	1800	200	50	8
• SUSPCION IDENT.	100	90	D	D	2910	220	100	0
• STATUS REQUEST	100	100	D	D	0	300	250	0
RECORD KEEPING								
• SYSTEM STATUS MAINT	350	800	D	D	8820	400	300	0
• DATA RECORDING	-	-	-	-	-	-	-	-
• FAULT CONDITIONS	700	280	D	D	D	300	100	60
• MODE CHANGES	10	20	3	T	2552	-	-	0
FAULT VERIF. & HANDLING								
• FAULT VERFICATION	20	30	D	D	-	100	30	0
• ACTIVE FAULT ISOL.	100	350	D	D	3000	3000	1000	0
• FAULT CORRELATION	100	450	D	D	-	1000	1000	0
• FAILURE THRE. TEST	20	30	D	D	2600	75	20	
• EMERGENCY HANDLING	320	60	D	D	-	30	-	30
FAULT TOLERANCE I/F								
• FAULT STATUS	100	100	D	D	D	100	50	30
• RECONFIGURATION VERFICIATION	30	50	D	D	D	200	50	20
ITM COMPONENT CONTROL	100	500	40	0	0	0	0	0
TOTALS	7430 W	5670 W	2260 OPS	401 WPS	27 kw	176 OPS*	- 71 WPS*	27 K * **

D= ON DEMAND

13100 WORDS

* ASSUME 1 FAULT/MINUTE; 1 PILOT DIRECTIVE/2 MINUTES
** ASSUME 5 HOUR MISSION

TABLE 20. ITM IMPACT ON ADVANCED SYSTEM

Resource	Impact	Percentage of total
Master processor memory	14,234 words	5.5
Master processor throughput	3,000 ops	0.3
Moderate speed (1553B) bus loading	571.2 wps	3*
System mass memory storage	70.2K words	1

*Assuming 1 MHz and 60% efficiency.

6.6 Key Issues Affecting the Architecture Development

In the course of developing the ITM system requirements, several key issues were identified that have an impact on the development of the avionics architecture. A discussion of these issues and their impact on the architecture development is provided in the following paragraphs.

a. Issue 1—response to a power-up failure of mission processors. In the preflight environment there is a requirement for the mission processors to perform a comprehensive power-up self test. This is to ensure that the ITM software is not loaded into a defective processor.

   Since the ITM control software is not loaded into any processor at the point of power-up, it is necessary to provide for the detection of a processor fault and the subsequent load of ITM software into the remaining operating processors. This will have an impact on the processor design or the processor interface to the controls and displays.

   The processor design will have to provide for a mechanism for the self-test function to communicate the detection of a fault to the load function (wherever it's mechanized) in order to initiate a load into an alternate processor. If the pilot is involved in the load process, the controls and displays will have to provide the capability of displaying a processor fault without the presence of operating ITM software.

b. Issue 2—testing of mission processor bus interfaces. In the preflight environment, there is a requirement to verify that all backup redundant resources will be operational when needed. This has particular impact for the mission processors and their interface to the data buses.

In the multibus architecture, each processor may interface with several buses and the bus interface unit will have built-in capability to act as either a bus master or a bus remote terminal. This means that if an interface is intended to be used as a remote terminal but may be called upon to act as the bus master when the original bus master fails, then it must be tested during preflight to ensure it can function as a bus master.

This can be accomplished in either of two ways. The first is to require that each device that is a backup bus master be switched to become a bus master and test its functions by communicating with other devices on the bus. The second is to provide a comprehensive built-in self-test of the bus interface unit that tests all interface functions, even hardware and firmware that is unique to either a remote mode or a master mode. The test must be able to be initiated either by the bus master if the device is a remote terminal or by the host processor if the device is a bus master.

c. Issue 3—redundancy of system mass memory. Use of the system mass memory is critical to the operation of the ITM. In preflight and inflight, the primary purpose of ITM is to detect faults and provide data that assists in reconfiguration of the system in order to complete the mission.

In postflight the primary purpose s to isolate the faults that have occurred in preflight and inflight. This is accomplished by recording of test data and associated status and environmental data on the SMM, then recalling the data during postflight diagnostic testing.

If a failure were to occur that would cause the SMM not to record data or not to recall the data, ITM would be incapable of performing as required. One solution is to provide some fault tolerant recording capability for ITM fault data. The alternative is to tolerate the reduction of ITM capability caused by the loss of SMM.

The ITM capability compromised is the ability to isolate faults that cannot be duplicated in the normal ground test environment. Faults that are reproducible on the ground can still be isolated after the SMM has been repaired (needed since it is used with the postflight test program).

6.7 MIU Design Requirements

The MIU is part of the maintenance interface during ground maintenance. It provides a central collection and display point for all maintenance and service requirements, both avionic and nonavionic. It needs to be located so that it can provide access from the ground without requiring the removal of any parcels.

During postflight, the maintenance technician reviews a sequence of messages on the MIU to identify which LRU's need to be replaced and what nonavionics maintenance actions are required. Messages for the avionics are stored as a result of the diagnostics performed during postflight testing, and messages for nonavionics are stored as a result of changes in nonavionics BIT acquired over the multiplex bus. The maintenance technician determines what repairs or service is required by stepping through the sequence of displayed messages.

Table 21 lists the key differences between the F-18 maintenance monitor panel and the MIU as specified in the ITM system specification.

6.8 Physical and Environmental

Since the MIU is the only hardware unique to ITM, the physical and environmental requirements relate only to the MIU. The requirements in the ITM system specification are a standard set of requirements for electronic equipment exposed to the environment expected in the nose wheel well.

AD-A138 587 INTEGRATED TESTING AND MAINTENANCE TECHNOLOGIES(U) 3/4
BOEING AEROSPACE CO SEATTLE WA R O DENNEY ET AL.
DEC 83 AFWAL-TR-83-1183 F33615-81-C-1517

UNCLASSIFIED F/G 5/1 NL

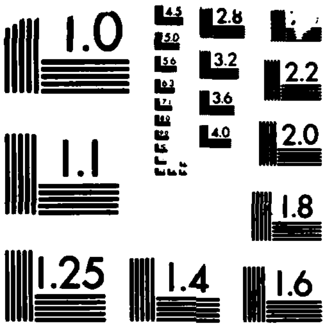

1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.2
2.0
1.1
1.8
1.25
1.4
1.6

TABLE 21. MIU REQUIREMENTS

Item	F-18 MMP	ITM requirement	Rationale
Display	3-digit code	3-digit number plus 16	Include identifying message number and alphanumeric charactersmake messages readable without looking up codes
Message length	N/A	32 characters	Allow complete message without resorting to codes
Number of messages	64	128	Provide for additional avionics
Order of messages	Stored in chronological order	OTP can recall, delete, reorder the messages	Permits prioritization of maintenance activities

7.0 COST ANALYSIS

Specific cost analysis cannot be provided without defining the system and its operation and support concept. This section will, however, give some insight into the scope of cost savings that can be provided by implementing the ITM concepts. As was previously stated, incorporation of ITM into a system will increase acquisition cost but decrease operation and support cost.

Figure 34 illustrates the distribution and time relationship of the contributions to the life cycle cost of a typical system. Figure 35 further breaks down the operation and maintenance costs. The significant cost is the repair labor cost, 32% of the system life cycle cost. It is in this area that ITM can reduce costs. The question is how much and to what extent does this savings exceed the cost of development.

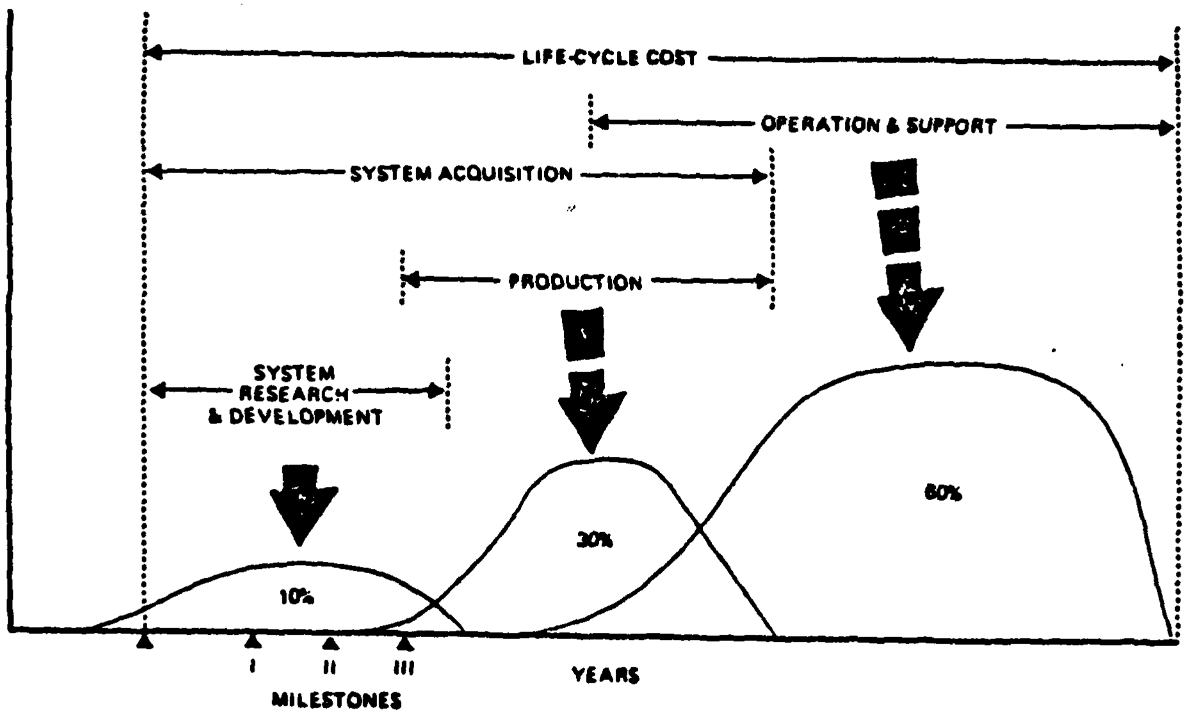
Figure 36 presents an estimate of the life cycle cost savings that could be realized by implementation of ITM. The first column lists the contribution of each element to the total life cycle cost. The second column provides an estimate of the impact of ITM. These are cost deltas to current implementations of onboard test and maintenance implementations. An attempt has been made to be very conservative. The various contributions are discussed below.

a. R&D design—typical systems devote 10% to 15% of the hardware design to BIT and less than 10% of the system software to test and maintenance. The estimate assumes an additional 10% contribution to the system software design effort for ITM.

b. Production—again, 10% to 15% of the cost of hardware production of a typical system is devoted to BIT. The estimate of 5% additional production cost is assumed to be half devoted to additional BIT requirements (i.e., temperature reporting) and half to the cost of additional computing resources for the ITM software.

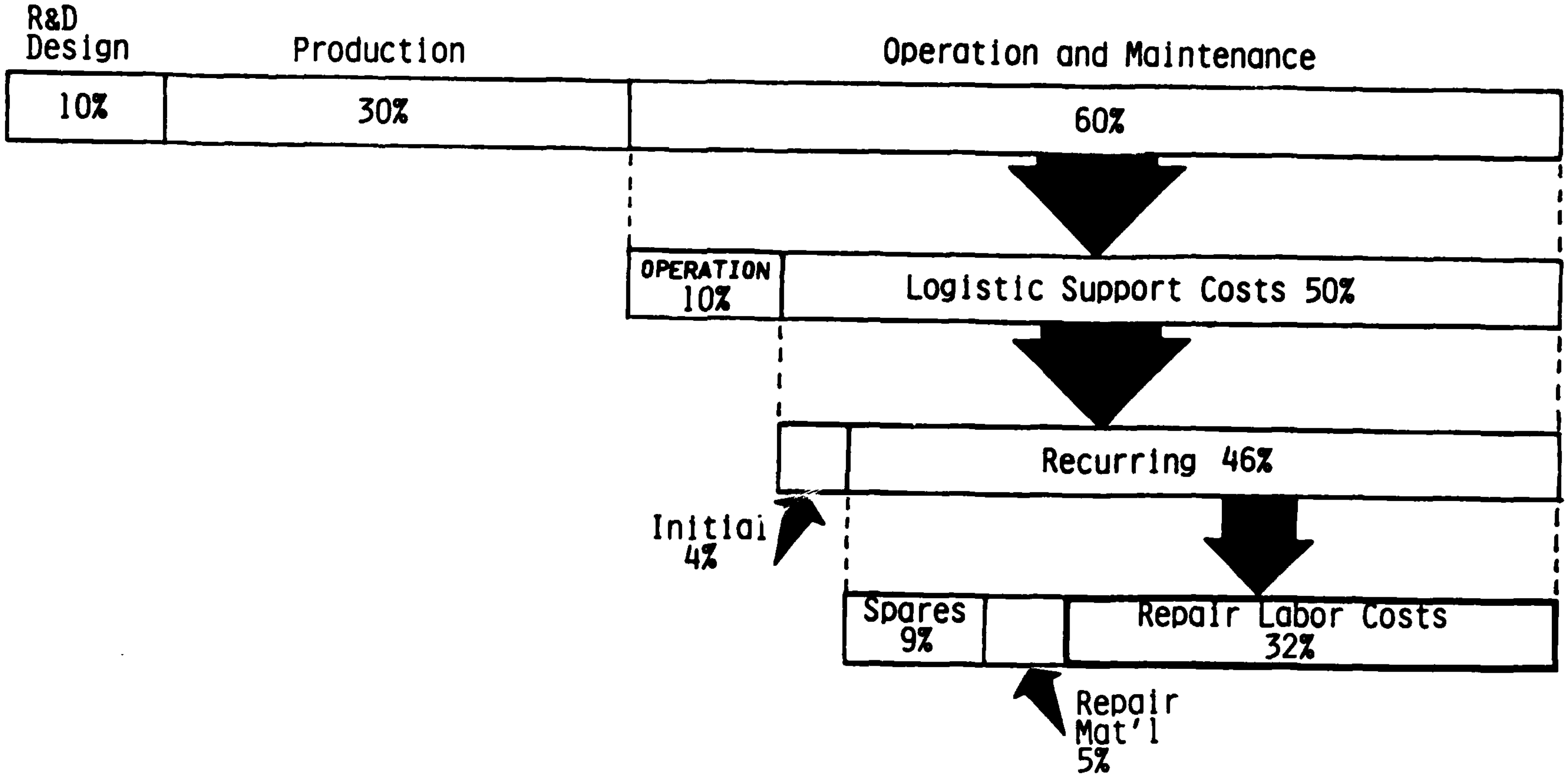
c. Repair labor—the 13.1% is the reduction of labor devoted to pursuing CND's and RTOK's.

d. Spares and repair material—this reflects the reduction in spares as a result in the reduction of CND's and RTOK's.



Source: Program Management for Functional Managers, The Defense Systems Management College, Fort Belvoir,

Figure 34. Typical System Life-Cycle Cost



From Program Management for Functional Managers,
The Defense Systems Management College, Ft. Belvoir, Virginia

Figure 35. Typical System Life-Cycle Cost Breakdown

LIFE CYCLE PHASE	% OF SYSTEM LIFE-CYCLE COST	X	ROUGH ESTIMATE OF ITM COST IMPACT	=	IMPACT OF ITM ON LIFE-CYCLE COST
● <u>R&D Design</u>	10%		Up 10%		Up 1.0%
● <u>Production</u>	30%		Up 5%		Up 1.5%
● <u>Operation & Maintenance</u>	(60%)				
Repair Labor Costs	32%		Down 13.1%		Down 4.2%
Spares & Repair Material	14%		Down 15%		Down 2.1%
Operation	10%		Down 15%		Down 1.5%
·Initial Logistics Support	4%		—		—
	100%				Down 5.3%

Figure 36. Potential ITM Cost Impact

e. Operation—this reflects the reduction in operation expenses as a result of reduced labor and reduced spares.

f. The final column combines the factors to yield the total cost impact of ITM. The result is an estimated savings of 5.4% of the total life cycle costs.

8.0 ARTIFICIAL INTELLIGENCE APPLICATIONS TO ITM

8.1 PROSPECTS OF ARTIFICIAL INTELLIGENCE IN AVIONICS SYSTEMS

Recently, much attention has been focused on artificial intelligence (AI) by industry, the military, and the popular press. The general feeling is that, given the current advances in hardware technology, AI is finally at the stage where its techniques can advance from academia into useful real-world applications. There are several definitions of AI, but the one used in this report is—

> Artificial intelligence is the study of how to make computers do things which, at the moment, people do better.

Because much of ITM results in automation of current human activity on the flightline, AI was studied to determine what the potential benefit of its application to testing and maintenance would be.

8.2 AI STUDY OBJECTIVES AND APPROACH

There were two objectives in the study of AI techniques. First, to determine if any of these techniques could be effectively applied to ITM. The second, to develop design concepts for suitable applications of AI to ITM. The scope of the study was limited to application of AI techniques to onboard testing at or above the LRU level.

The approach taken to achieving these objectives was to—

a. Survey the AI field to determine what techniques are used and how they are applied. This task included a survey of literature, interviews with Boeing AI experts, and contact with other experts.

b. Evaluate the potential AI applications to ITM. This task consisted of reviewing the ITM system specification to identify features and processes that might have potential for application of AI and associating AI techniques with these ITM features. A candidate AI task was then selected for development.

c. Develop design concepts for the selected AI application. Here, tasks performed included interface definition, problem representation, software structuring, sizing, and timing.

8.3 ARTIFICIAL INTELLIGENCE TECHNIQUES STUDIED

Although the definition of AI is not generally agreed upon, the fields that comprise it are. In order to determine how AI techniques could best be applied to integrated testing and maintenance, each field was surveyed. They are described, for instance in Nilsson's Principles of Artificial Intelligence(11). These fields are described below. All are still in their developmental stages, although there have been promising results.

Natural language processing—the branch of AI that attempts to create machines with humanlike communication skills. This field must borrow heavily from linguistics and other sciences to understand how humans communicate.

Natural language processing programs are usually used as an interface between a nontechnical system operator and a complex computer system. For example, the natural language interface could be used as an interface to an intelligent data retrieval system. This would allow the ITM system to respond to operator-initiated questions such as, "What was the internal temperature when the ICNIA signal processor failed self-test"? with answers such as, "The avergage temperature was 45°C. Actual temperatures at the time of the three failures were 68°C, 25°C, 42°C. Specified high temperature for that box is 60°C". Similarly, a natural-language interface could be used in conjunction with an expert system or an intelligent computer-aided instruction (ICAI) system.

In future systems, a natural language interface would add to pilot and crew performance. However, the natural language capabilities could be used by many subsystems and should not, therefore, be developed as an ITM-unique function. They will be considered for application to ITM because of their potential for use in future systems.

Intelligent computer-aided instruction (ICAI) systems—used to efficiently instruct a student on a given subject. The systems consist of: (1) problem-solving expertise, which is the knowledge the system tries to impart to the student; (2) the student model, which monitors what the student does and does not know; and (3) tutoring strategies, which specify how the system presents material to the student (12). A good deal of interesting work has been done in the ICAI domain and can be found in the literature. ICAI techniques will be considered for ITM application.

Intelligent retrieval from data base—a means by which an avionics maintenance crewmember can acquire a faster understanding of fault characteristics. The retrieval scheme might, for instance, contain a set of simple correlation pairs that are analyzed during postflight. These pairs might check a given type of fault against flight mode, acceleration, temperature, etc., and the system would notify the crewmember of any recognized correlations. Additionally, the system could contain more complex correlations such as, if conditions A, B, and C occur within 10 min, then fault X probably is causing the failures.

Such a system could also have built-in improvability. If the avionics crewman spotted any system characteristics that were triggered by a given failure, he could add that knowledge in the form of a correlation test. The test would subsequently be checked as part of the normal postflight correlation exercises.

Intelligent retrival techniques could also use the ITM mission-to-mission fault data catalog. This catalog contains information defining the recent and long-term failure rates for the system. Use of this catalog would support correlation production rules that could determine whether a fault was occuring more often that its normal background rate.

Many of the ITM postflight functions require data acquisition similar to that of the intelligent retrieval systems; therefore, they will be considered for application to ITM. If an intelligent retrieval approach is taken, it would be necessary to determine the costs and benefits relative to the alternative, conventional approach.

Expert systems—store the knowledge of an expert. The system is able to retrieve and process the stored knowledge to perform such functions as diagnosis, monitoring, prediction, and planning. Currently, all expert systems are "rule based"; that is, the knowledge is stored in the form of if-then or situation-action rules. These rules (also called production rules) form a network of inferences that are used to perform the expert functions.

An expert program can be loosely divided into three sections: a knowledge base, a global data base, and a control mechanism. The knowledge base contains rules used in the domain of expertise. The global data base contains the input data, data generated by the rules, and the current state of the system. The control mechanism contains the strategy of rule application and rule search.

There are many techniques for implementing expert systems. However, there is no standard approach to choosing and applying these techniques, and development of expert systems remains somewhat of an art form. Basically, definition of an expert system consists of two parts: defining the direction of the inference structure, and defining the search strategy to be employed.

The direction of inference shows the way problems are solved by the expert system. For instance, a forward chaining, or data driven, system begins with current data and works toward a goal by chaining through the rules. A backward chaining, or model driven, system is used when a goal or a set of possible goals is known. The system works backwards from the "then" to the "if" parts of the rules to find the set of initial conditions which could lead to the goal. Some systems use a combination of forward and backward chaining to find a path through the inference network.

One expert system that was developed for medical diagnosis (MYCIN), uses a backward chain search (13). First, the program generates a set of hypothesized diseases for the patient's input data. For each of these diseases, the computer traces backward through the rules from causes to subcauses until it finds the disease with symptoms that best matches the patient's symptoms. MYCIN has a special feature which allows rule data to be stored along with a certainty factor, this allows alternative solutions to be reviewed in a probabilistic manner.

As an example, an avionic diagnostic system might operate in a similar manner. The input data or "symptoms" would be a record of faults and fault conditions that occurred during flight. Additionally, the program could acquire more input data by resuming tests or by requesting crew inputs. The knowledge base would contain rules of cause and effect relationships in electronic systems. For instance—

If bus communication fails, then the—

a. Transmitter box was faulty, or
b. Receiver box was faulty, or

c. A bus-interfaced box broadcasted noise, or

d. Background EMI was present.

The corresponding global data base would provide information such as—

a. Bus-interfaced boxes are—
   1. Mission Processor 1
   2. Displays Processor
   3. System Mass Memory
   4. ICNIA Data Processor
   5. etc.

b. Stores Launcher Unit has current suspicion index of 0.73

c. Temperature at time 1704 was 43°C

Several diagnostic expert systems have been built and are used today. They are summarized in appendix F. Expert systems techniques will be considered for application to ITM.

Theorem proving—this branch of AI attempts to produce programs that can use given logical concept to prove new ones. This is not immediately applicable to the testing and maintenance field because the real world concepts and rules are not well defined and the input data is noisy. Current theorem proving systems using predicate calculus and propositional logic are not readily adaptable to this scenario. Future developments in the field of fuzzy logic may make theorem proving more applicable.

Robotics—the field of robotics is associated with electromechanical operations and not applicable to diagnostic tasks, so it was not considered for ITM applications.

Automatic programming—these techniques would allow for software routines to be written by the program that would use them. While this offers some interesting possibilities, it is not applicable to the highly controlled software environment of the production fighter squadron.

Combinatorial and scheduling problems—these problems, which are usually solved best by operations research techniques, can sometimes be solved through use of AI principles. However, there are no ITM problems of this type that are complex enough to require innovative solutions.

Perception problems—these were not addressed because they are most applicable to machine vision and sensor input recognition.

8.4 SELECTION OF AN AI APPLICATION FOR DEVELOPMENT

The following describes various ITM processing capabilities with potential for application in one or more of the various fields of AI. In the descriptions, those features that make these ITM processes particularly suitable for AI applications are emphasized.

a. Filter false alarms and intermittent faults (preflight and inflight).

The current conventional approach is to establish time or number thresholds for failure reports—usually the same threshold for all failures. The ITM specification establishes time, number, and frequency thresholds for each failure report. The problem is, distinguishing false alarms from intermittent failures and determining when intermittent failures are affecting performance is a function of numerous other factors: environment, modes of operation, etc. The conditions and interrelationships are many, and the decision mechanisms are not clear cut. This is a problem that looks suitable for an expert system. The system could, for each failure indicated by BIT (if it is not a hard failure), use previous data and information about associated operating conditions to determine if the indication is a false alarm, and if not, determine if the indicated failure is signifiant enough to warrant corrective action.

b. Integrate data to determine fault source (preflight and inflight).

This is a diagnostic problem, but does not require the level of isolation of postflight diagnosis. As with any diagnosis problem, there are a large number of solutions possible and, typically, limited or ambiguous data. The goal of preflight and inflight diagnosis is to isolate the failure only to the extent necessary to determine whether or not the failure can be circumvented by reconfiguration and what needs to be reconfigured. This usually is not a complicated process, but in the envisioned highly integrated avionics system, the source of the failure may be difficult to identify. This is another possible application of an expert system.

c. Determine funtional capability (preflight and inflight).

The ITM system is primarily a BIT data collection and processing system with supplemental system-level and functional tests. As such, test results are generally structure oriented rather than function oriented: The test results determine which box is bad but not necessarily which function is affected or to what extent. Determining functional capability requires making conclusions from an incomplete set of data, but conclusions can be made by considering all of the existing data and conditions when the test data were collected. An expert system could aid in this process.

d. Selective notification to pilot (inflight).

Failures are not to be reported to the pilot when unnecessary or deterimental to his work load. The ITM baseline concept is to have a matrix of failure catagories versus flight modes with indications of which failure report types to suppress during certain flight modes. An expert system could extend the effectiveness by determining the functions affected by the failure and considering the immediate and anticipated pilot and system actions in order to determine which failures to report and when.

e. Suspicion identification (inflight).

This capability allows the pilot to make entries into the ITM system when he suspects there is a problem. Since the pilot is not fluent in the language of the test system, the conventional approach would be to implement communication with a few key words or through the use of menus. Implementing this capability with a natural-language input would allow the pilot to communicate with the system in his own language.

f. Restoration of failed devices (inflight).

Since some failure indications can be intermittent, a failure that has been declared for a device can become less severe or disappear. The ITM specification requires continued testing to determine when a device tests fully operational again. The decision of when to use the device is a complicated decision that is a function of how long the device malfunctioning, what functions were affected, the criticality of the device or function, and what other conditions were present when the device was malfunctioning and when it was working. This problem appears suitable for an expert system.

g. Classification of faults (postflight).

This process involves analyzing the failure data recorded in flight and the data saved from mission to mission in order to determine which failure reports represent which conditions—hard failures, false alarms, transient faults and intermittent faults. The objective is to determine which failures need to be (or can be) repaired. This may be suitable for an intelligent retrieval from data base process or an expert system, or possibly a combination of both.

h. Associate intermittent faults with modes of operation, etc. (postflight).

To isolate intermittent faults, the indications must be associated with the conditions contributing to the failure. These may include modes of equipment operation, environmental conditions, flight phases, or other failures. The problem is to determine from a large number of possible conditions those that are specific contributors to the failure indication. An expert system may be suitable.

i. Diagnose faults (postflight).

In postflight diagnosis the objective is to isolate failures to an LRU. Typically there is a large solution space and limited or ambiguious data. In the case of intermittent faults it may be necessary to isolate failures using recorded data. An expert system can perform the diagnosis or be used by the maintenance technician in the diagnosis.

j. Interact with maintenance crew (postflight).

Inte^raction with the maintenance crew is an essential element of the postflight operation. This includes output of the diagnostic results, inputs of diagnostic information from the maintenance crew, and access to the failure data base. Elements of natural language processing, intelligent computer-aided instruction, and intelligent retrieval from data bases can be combined in a system with which the maintenance crew can easily communicate and can learn how the system diagnoses problems. In this way the maintenance crew gets on-the-job training.

k. Manual isolation (postflight).

To isolate and repair those problems the system cannot diagnose, the maintenance technician must be able to access all of the available test data and exercise system tests. Since the maintenance technician is requesting and receiving information, natural language processing features are applicable, and retrieval of appropriate data from the large data base may be suitable for the process of intelligent retrieval from data bases.

l. Download maintenance data (postflight).

The capability specified for ITM is to download into the LRU relevant failure, environmental, and operational data to assist depot maintenance and failure history tracking. It is inappropriate to store all data that have possible association with the fault. Therefore, the system must select which data to download. Because failure mechanisms are complex, an expert system has potential for selecting the data for each failure situation.

m. Mission-to-mission catalog (postflight).

Failure data for uncorrected failures are saved from flight to flight to be used in determining capability in subsequent flights and to use in tracking and diagnosing intermittent faults. In addition to the failure data, associated operating and environmental data must be saved when appropriate. Retrieving the appropriate data and storing it in a useful way is a potential task for an expert system using intelligent retrieval from a data base.

In following the ITM task approach described in section 8.2, the proposed AI applications were pared down to one. The objective was to find the best application to develop.

A matrix of ITM processes and associated AI techniques is presented in table 22. The matrix was reduced as follows:

a. Use of natural language processing was deleted from consideration because there was nothing unique to ITM in the application of the technology (except vocabulary); the user would have familiarity with the system and could use the system's language; and the payoff is small in relation to the large amount of computer resources required.

b. Use of computer-aided instruction was deleted from consideration because training per se is inappropriate to the ITM application. ITM will still incorporate instruction capability, but it will be of the nature of instruction by example and query as opposed to the intelligent computer-aided instruction. In addition, this field is being worked in other AFWAL efforts, particularly at the Human Resources Laboratory.

c. The application to inflight functions was deleted from consideration because of the limited computing resources available to test and maintenance in flight.

The remaining candidate ITM functions were compared to the testing and maintenance problem areas to determine which had significant impact on the most problem areas. This narrowed the list down to—

a. Filtering false alarms and intermittents (preflight).
b. Classification of faults (postflight).
c. Associating faults with modes, etc. (postflight).
d. Diagnosing faults (postflight).

In evaluating these it was determined that—

a. Associating faults with operating modes, environmental conditions, flight conditions, and so forth, is a subtask for classifying faults as hard faults, intermittents, false alarms, etc.

TABLE 22. ITM PROCESSES VERSUS AI FIELD MATRIX

	Natural Language Processing	Intelligent Computer Aided Instruction	Intelligent Retrieval from Data Bases	Expert Systems
<u>Preflight</u>				
Filter false alarms and intermittent faults.				X
Integrate data to determine fault source.				X
Determine functional capability.				X
<u>Inflight</u>				
Filter false alarms and intermittent faults.				X
Integrate data to determine fault source.				X
Determine functional capability.				X
Selectively notify pilot.				X
Identify suspected failure.	X			
Restore failed devices.				X
<u>Postflight</u>				
Classify faults.			X	X
Associate intermittent faults with modes, etc.				X
Diagnose faults.				X
Interact with maintenance crew.	X	X	X	
Manually isolate fault.	X		X	X
Download maintenance data.			X	X
Catalog mission-to-mission failures.			X	X

b. Classifying faults is a key prerequisite to filtering false alarms and intermittent faults in preflight and to diagnosing faults in postflight.

c. Classifying faults benefits problem resolution more than diagnosing faults, because CND and RTOK rates are the result of lack of understanding and tolerance to false alarms and intermittent faults.

d. Classifying faults also supports filtering false alarms and intermittent faults inflight if computer resources are available for implementation.

In conclusion, the most beneficial application of AI to ITM would be an expert system to classify faults as hard faults, intermittent faults, false alarms, etc., using data recorded in flight. To further evaluate the suitability of applying an expert system to this ITM function, the application was evaluated with respect to a set of prerequisites for construction of a successful expert system as reported by Gervarter (14). This evaluation is summarized in table 23 and concludes that this application meets those prerequisites.

TABLE 23. SUITABILITY OF EXPERT SYSTEM TO CLASSIFICATION OF FAULTS

	Prerequisites for construction of a successful expert system	Evaluation of ITM fault classification application
1.	Must be at least one human expert acknowledged to perform the task well.[a]	The subsystem/LRU BIT designer or the system/subsystem test integrator is the appropriate expert.
2.	Source of knowledge must be: Special knowledge, judgement and experience.	Definitely requires special knowledge of subject and judgement weighed by experience for most of the problems that occur.
3.	Well-bounded domain of application.[a]	Bounded in the sense that the physical system is well bounded in failure modes. Note: domain may be extensive.
4.	Doesn't require only common sense.[b]	Expert needs experience with testing of the specific systems.
5.	Takes an expert a few minutes to a few hours.[b]	Most problems take less than a few hours, but problems the expert hasn't seen may take longer.
6.	Has an expert available and willing.[b]	This needs to be programmed into system development.
7.	Problem nontrivial but tractable.[c]	The problem is definitely nontrivial, but whether or not it is tractable requires further evaluation. The bulk of the problem is tractable.

a Duda
b Davis } reported by Gervarter (14)
c Hayes-Roth

8.5 DESCRIPTION OF SELECTED EXPERT SYSTEM DOMAIN

8.5.1 Selection of E-3 IFF Subsystem

Once it was determined that an expert system would perform fault classification functions, it was necessary to select an actual avionic subsystem to which it could be applied. After reviewing necessary criteria, the E-3 IFF subsystem was chosen from among several other candidates. Potential applications were nominated on the basis of having known problems with intermittents and false alarms, applicability to the fighter avionics concerns, and availability of inhouse expertise. The criteria used to rate the potential applications were (table 24)—

a. Expertise available—an expert on fault diagnosis for the particular subsystem would be needed to develop the rule base. Among the few with which we had adequate expertise, the IFF subsystem was best.

b. Subsystem generates intermittents and false alarms—this would ensure that a sufficient variety of data would be available and that the expert system would be targeted to an area where it was needed. All candidates qualified in this category.

c. Difficulty of solution—as stated previously, an expert system's domain should be "nontrivial, but tractable." It was felt that the E-3 radar problems were too complex an undertaking for this effort. Other candidates appeared to be suitable.

d. Fighter application—the subsystem selected should be similar in structure or function to subsystems on modern fighter aircraft. Although the IFF subsystem used on the E-3 is not identical to those found on a fighter aircraft, the functions performed are similar enough that the outputs of the expert system development can be used on future tactical fighters.

e. Operating data available—a large amount of operational data should be studied to determine the characteristics of the false alarms and intermittent faults produced by the subsystem. Boeing's development of the E-3 aircraft provided this test data for all of its subsystems.

8.5.2 Description of E-3 IFF

The E-3 IFF is a Mark X/XII SIF/IFF-type interrogator system. It provides digital data reports indicating range, azimuth, altitude, selective identification features (SIF) and other coded identity of targets equipped with compatible transponders. The interrogation system consists of an interrogator set, AN/APX-103, and antenna equipment. The interrogator system interfaces to the mission computer and to the onboard test system. It receives commands from the mission computer and returns target data. It also provides test data to the onboard test system.

The interrogator set comprises two radar receiver-transmitters, two power supplies, a radar target data processor, two interrogator computers and an rf transmission line switch interconnected as shown in figure 37. The radar target data processor receives interrogation commands from the mission computer, codes the interrogations, and sends the interrogation to the active receiver-transmitter. The receiver-transmitter transmits the interrogation, receives the response and sends the response to the radar target data processor. The radar target data processor decodes the response and sends the target data to the mission computer. The interrogator computer is used by the radar target data processor for mode 4 interrogations. The radar target data processor establishes which receiver-transmitter and interrogation computer is active and switches the active receiver-transmitter to the antenna.

The modes of operation of the interrogation system are standby, passive, active and loop-test. When active, the modes are—

a. SIF modes:

  (1) Mode 1 - military; group identification.
  (2) Mode 2 - military; aircraft identification.
  (3) Mode 3/A - military/civil; identification, information or tracking.
  (4) Mode C - military/civil; altitude.
  (5) Two mode interlace - any two of the above.
  (6) Three mode interlace - any three of (1) through (4).

b. Supermode - mode 4 (classified) interlaced with any SIF mode, (1) through (6).

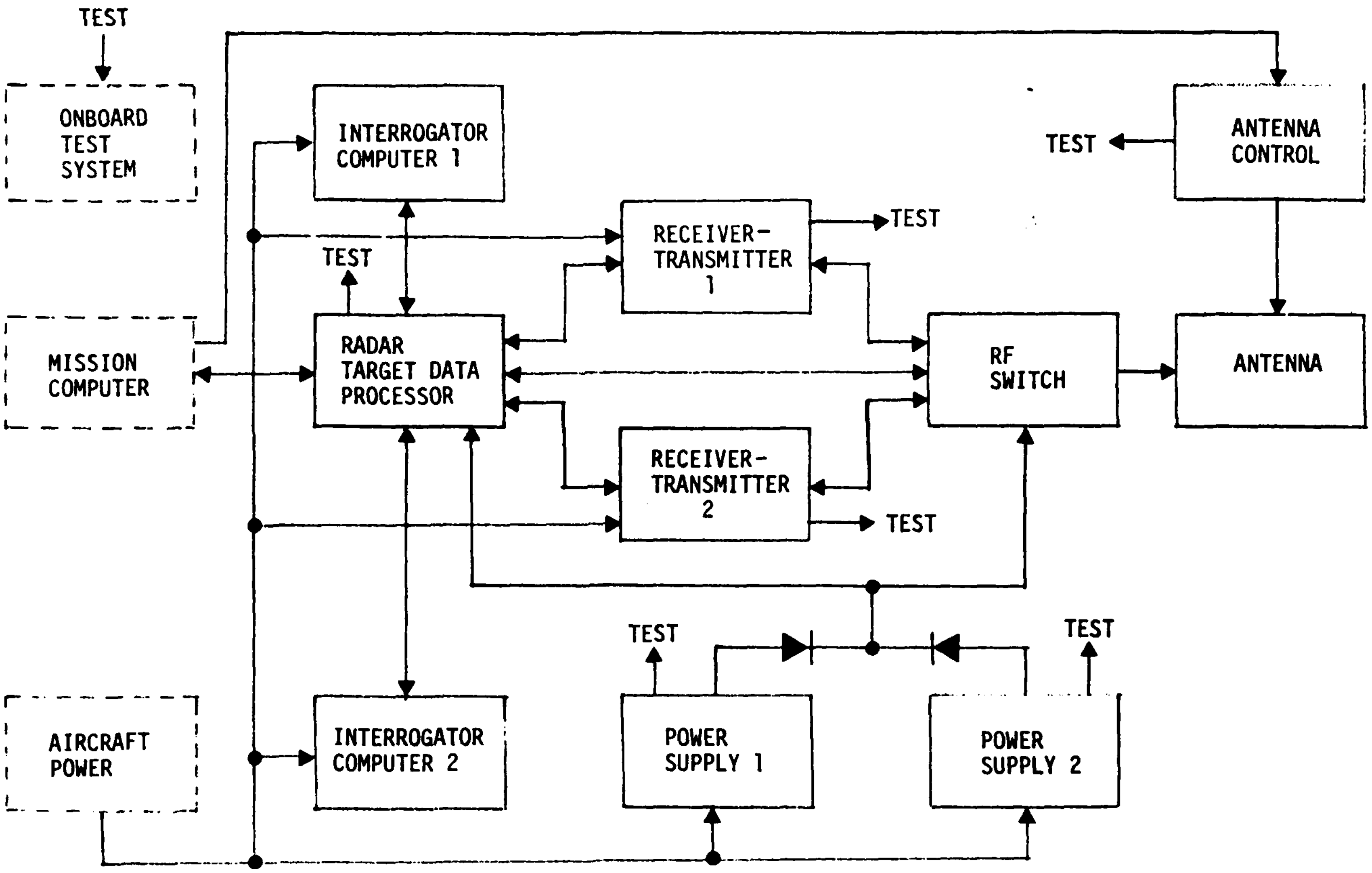


Figure 37. Interrogator System Block Diagram

The modes of the various components of the interrogator system are—

a. Radar target data processor (same as interrogator system, above).

b. Receiver/transmitter (standby, active).

c. Interrogation computer (standby, active (used only with mode 4))

d. Power supply (on only).

e. RF switch (on only).

f. Antenna control (on only)

8.5.3 Test Data for E-3 IFF

The E-3 IFF system provides test data to the onboard test function in the form of testpoint information that is collected periodically (every 6 to 10 seconds) by the test functon. The test data are in one of two forms—discrete or analog. The discrete signal is a go/no-go indication that can be the result of a test ranging in complexity from a very simple to a very complicated digital diagnostic test. The analog signal is a dc voltage that represents the result of an analog test. This voltage is compared to fail limits in the onboard test software. The number of tests of each type for each LRU is—

	Target data processor	Receiver/ transmitter	Power supply	Antenna control
Analog	2	8	10	0
Discrete	108	6	0	2

Test data for the interrogator computers and the rf transmission line switch are acquired by the radar target data processor, then sent to the onboard test system.

All the data specified for ITM, except LRU temperature, are available for the E-3 IFF. To be useful in checking out the fault classification expert system during development, a data reduction program would be required to process the recorded E-3 mission data. This program would extract the fault data and operating mode data from the tapes and format it for use by the expert system. Simulated temperature data should be inserted into the processed data for completeness.

In the process of developing the initial set of rules, flight test data were examined using existing E-3 recording tape processing facilities. It was apparent that the thousands of hours of data would have to be reduced to find the significant cases out of all the typical failure reports.

8.5.4 Test Characteristics for E-3 IFF

The design and implementation of the E-3 IFF is typical of recent avionic designs. As such, it experiences typical distribution of hard faults, transient faults, and intermittent faults. The false alarms caused by errors in the test design, while unique to the E-3 IFF, are also typical of the false alarm problems of other avionic systems.

8.5.5 Fault Classification Criteria

At the beginning of postflight, the inflight test record shows what faults occurred and what actions were taken. If a fault recorded in flight can be verified in postflight testing, it can be classified as a hard fault.

Intermittent and transient faults are those faults which do not constantly occur. They differ in that an intermittent fault reflects actual internal failure (this includes out-of-specification components) in a unit, whereas a transient fault does not. So, if an intermittent fault occurs, there is a higher probability that that fault will occur again, i.e.;

$$P(F_t / F_{t-a}) \quad P(F_t)$$

Where F_t is a fault at time t, a is a time lag, and $P(F_t / F_{t-a})$ is the conditional probability that F_t is true, given F_{t-a}.

For a transient fault, there is no dependence on previous faults since the causes of the fault are outside the system; i.e.,

$$P(F_t / F_{t-a}) = P(F_t)$$

From an observational standpoint, it is often difficult to tell the difference between hard, intermittent, transient faults, and false alarms. Some of the phenomena contributing to this are—

a. A hard fault might appear to be intermittent because the fault occurs only when the equipment is in a certain operating mode. A hard fault always fails in the given mode; however, it behaves like an intermittent fault, because the equipment works when it is not in the faulty mode.

b. A transient fault might appear to be an intermittent fault because of frequent environmental stress. In that instance, there would be no need to remove the faulty box, because no internal failure has been incurred. In order to avoid unnecessary removal, fault reports should be correlated with environmental factors to find hidden causes.

Classification of faults by the expert system requires correlation of inflight data with fault occurrences. Rules determine what data is used and how it is used. The general classification of the faults will be an input to the isolation diagnostics phase. A hard fault should be isolated to an LRU and the box removed. Intermittent faults should be isolated, but repair may or may not be necessary, depending on the frequency and severity of the fault. Transient faults might be isolated to gain insight into equipment performance but removal is not necessary.

TABLE 24. EXPERT SYSTEM CANDIATE SELECTION

Candidate	Expertise available	Subsystem generates inter-mittents and false alarms	Difficulty solution	Fighter application	Operating data available
E-3 IFF	Yes	Yes	Moderate	Similar function incorporated	Yes
E-3 radar	No	Yes	Too complex for this effort	Yes, most troublesome fighter subsystem.	Yes
E-3 displays or data processor	Yes	Yes	Moderate	Little similarity to fighter	Yes
E-3 communica-tions or navigation	No	Yes	Moderate	Yes	Yes
F-14 radar	Yes, but only with design phase	Yes	Moderate	Yes	No
Other fielded fighter systems	No	Yes	Varied	Yes	No

8.6 EXPERT SYSTEM IMPLEMENTATION

8.6.1 Description of Terms

The expert systems field is relatively new, so there is not yet a standard nomenclature. The same concept may have many names. Some of the terms used in this study are clarified in the following paragraphs.

An expert's knowledge could be conceptualized as a group of branches and nodes (figure 38). In this model, each node represents a fact or piece of data about the system the expert analyzes. Each collection of branches leading to a node can be considered a "rule" for deriving the fact or data associated with that node. Typically, the rules are logical, but complex, combinations of data.

The expert might begin at the left side of the graph with the basic data and would then draw conclusions from the data (i.e., move to nodes that are closer to the goals) by applying the rules. This process would continue until one of the goals was reached (which represents the solution to the problem being analyzed). Of course, an expert would have a much larger body of knowledge (branches and nodes) than the one shown in the figure. Some of the terms associated with the above process are defined as follows:

Problem states—the state is defined by which nodes have been reached; each state defines where the system is in the problem solving process.

Moves—a move is the attainment of a more advanced (farther right) state based on given or derived information; the application of a rule.

Goals—these are the possible final states of the system. When a goal is reached, a useful conclusion is drawn from the data.

Subgoals—any of the more important intermediate problem states can be considered a subgoal.

Goal condition—a set of values that can be tested to determine whether a goal has been reached.

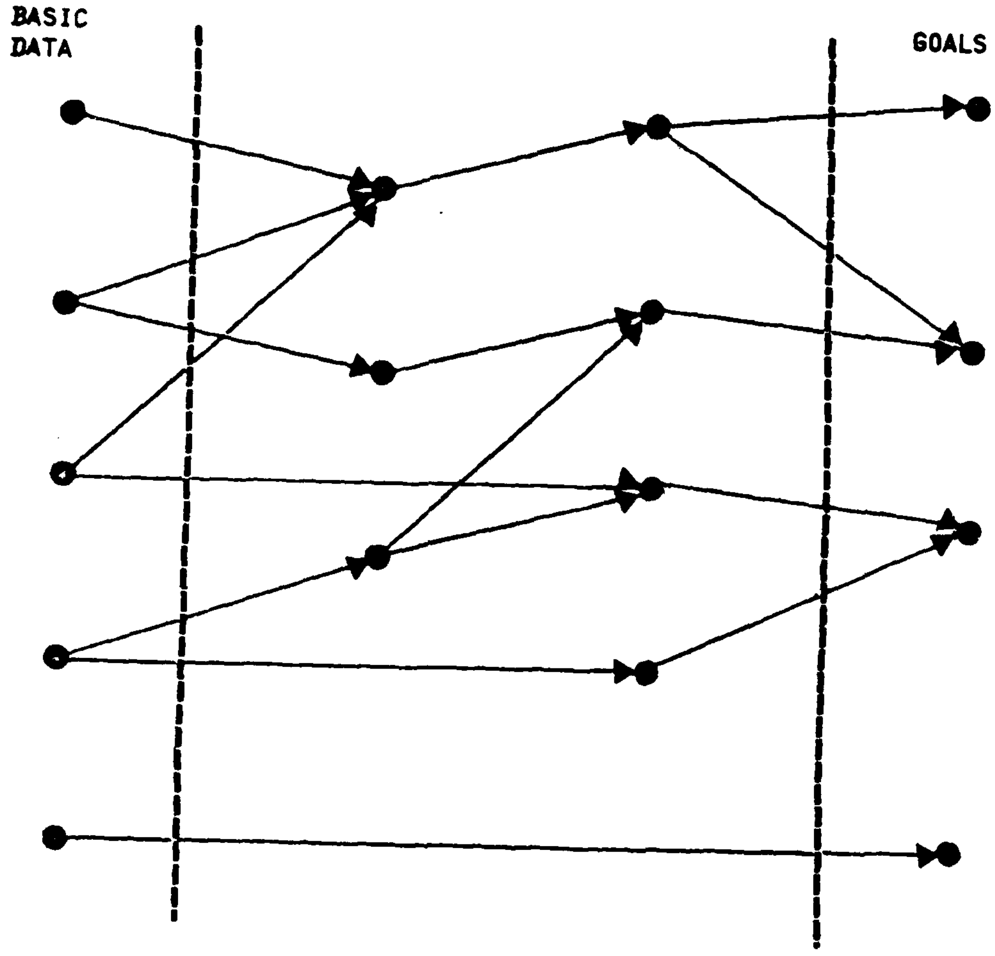


Figure 38. Expert Knowledge Conceptualization

8.6.2 Problem Representation

The problem representation was developed for the expert system. The problem has been broken down into six phases as shown in figure 39. Phases 1, 2, and 6 implement the interfaces with the fault data and the Postflight Operational Test Program (POTP). Phases 3, 4, and 5 constitute the main body of the expert system.

The primary goals of the expert system are the classification of faults (hard, intermittent, transient, false alarm, etc.), and the discovery of any operational dependencies of the fault (e.g., LRU mode, temperature). This data is used in the next phase of ITM processing to produce more accurate fault isolation.

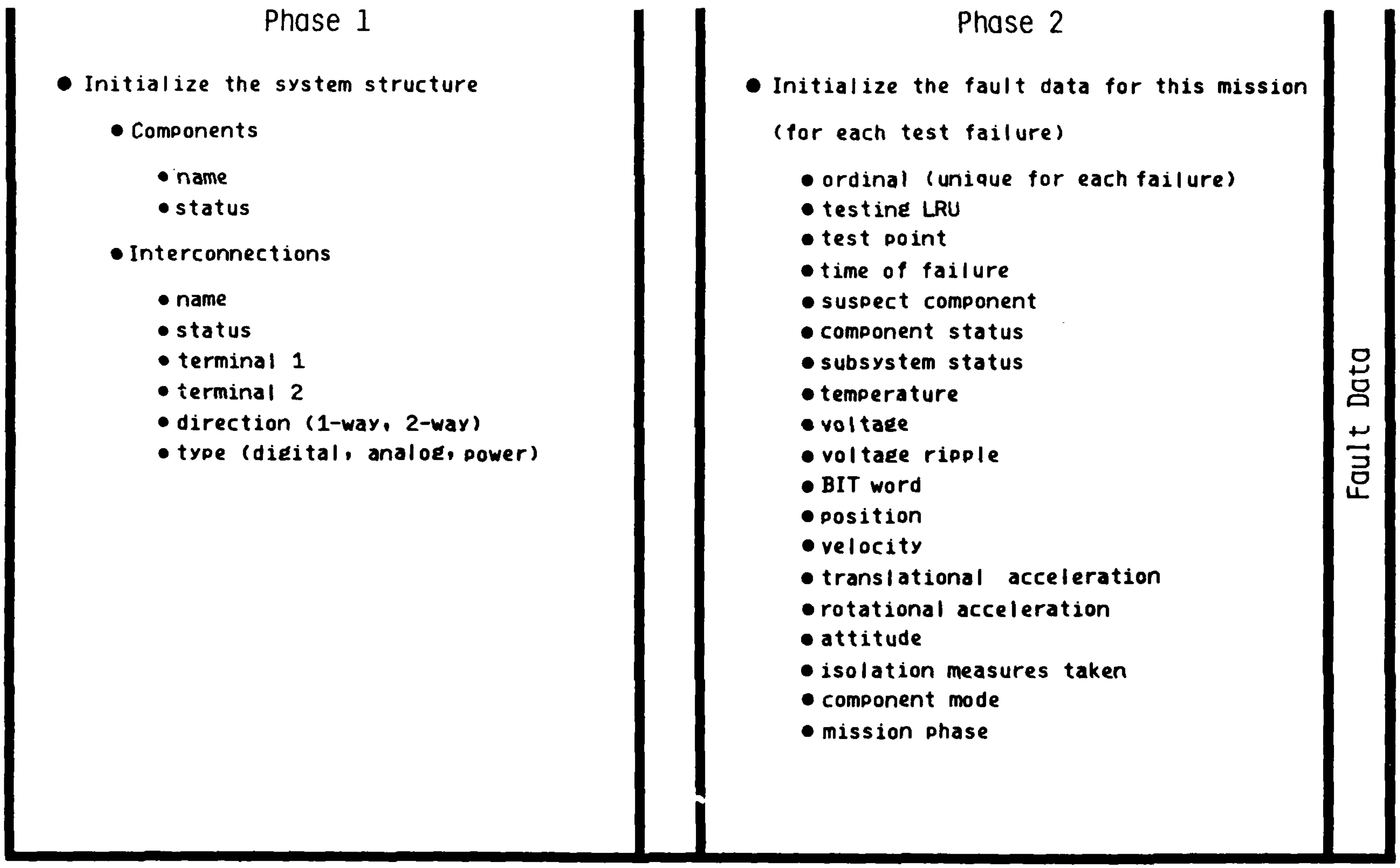
Grouping faults together by cause is a major subgoal of the expert system. Each fault (test failure) is a symptom of some failure in the hardware. Often, a single cause of hardware failure can produce many test failures. An expert analyzes groups of test failures rather than individual occurrences to diagnose fault causes.

The phases of the system are:

a. Initialization of the system architecture. During this phase, the components and the interconnect sections of the global data base are initialized. The rules will be written so that components may be added or deleted from the data base. The interconnects (including digital, analog, and power) will also be changeable.

b. Initialization of fault data for the mission. The fault data for the last completed mission is loaded into the data base. This data will include time of failure, test that failed, environmental data, and operational data.

c. Preliminary grouping. The initial action of the expert classifier will be to make a preliminary grouping of the faults. General rules of thumb are (1) all occurrences of the same test failure have the same cause, and (2) all tests failing at the same time have the same cause (exceptions to these rules can be built into the rule base). For the simple case where only one test failed (repeatedly), we can make the assumption that there is only one cause, and thus only one group. In other cases faults must be grouped according to time dependencies as well as repeated occurrences. At the end of this phase, each fault will be grouped with other faults



Phase 1

- Initialize the system structure
  - Components
    - name
    - status
  - Interconnections
    - name
    - status
    - terminal 1
    - terminal 2
    - direction (1-way, 2-way)
    - type (digital, analog, power)

Phase 2

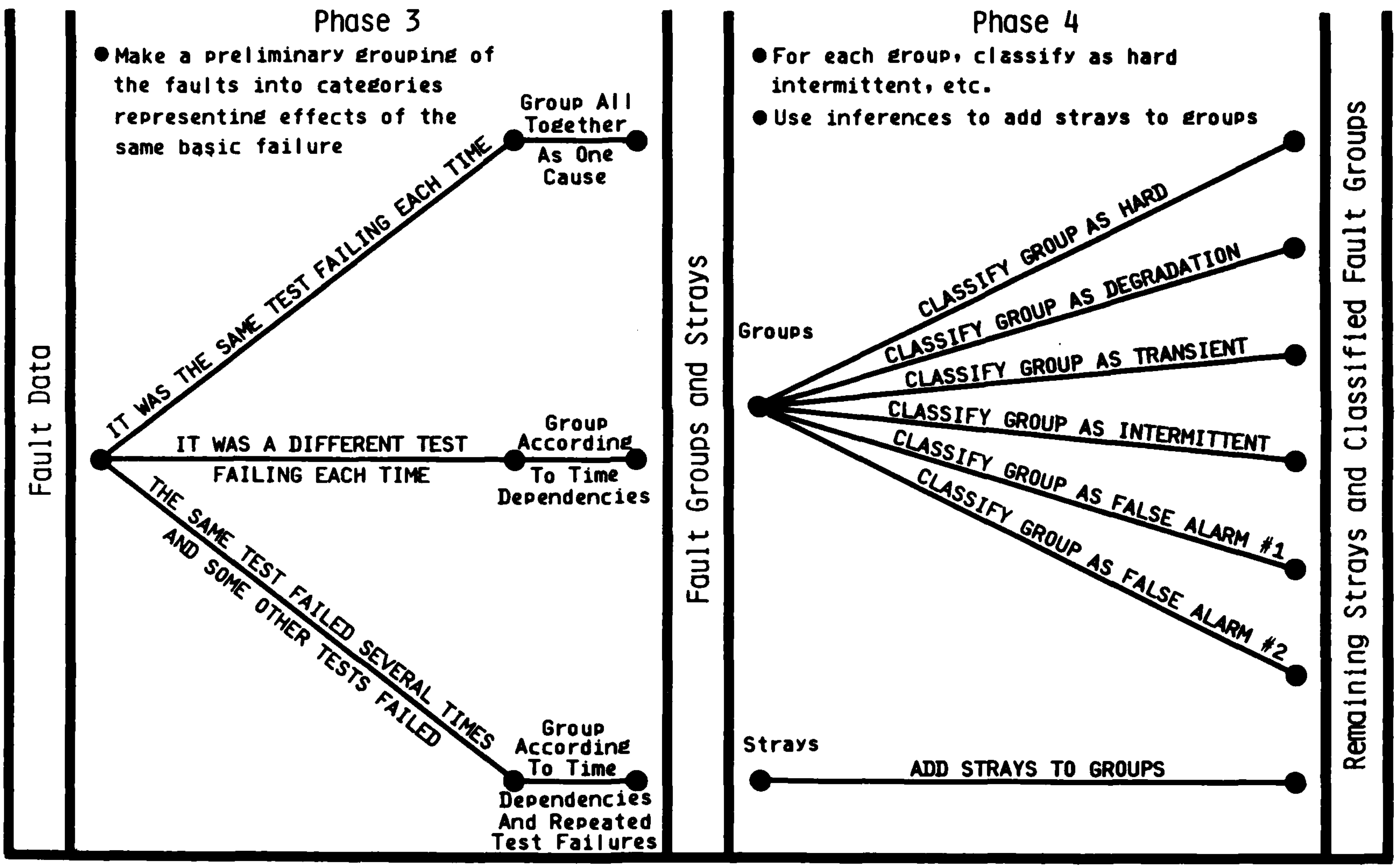
- Initialize the fault data for this mission

  (for each test failure)
    - ordinal (unique for each failure)
    - testing LRU
    - test point
    - time of failure
    - suspect component
    - component status
    - subsystem status
    - temperature
    - voltage
    - voltage ripple
    - BIT word
    - position
    - velocity
    - translational acceleration
    - rotational acceleration
    - attitude
    - isolation measures taken
    - component mode
    - mission phase

Fault Data

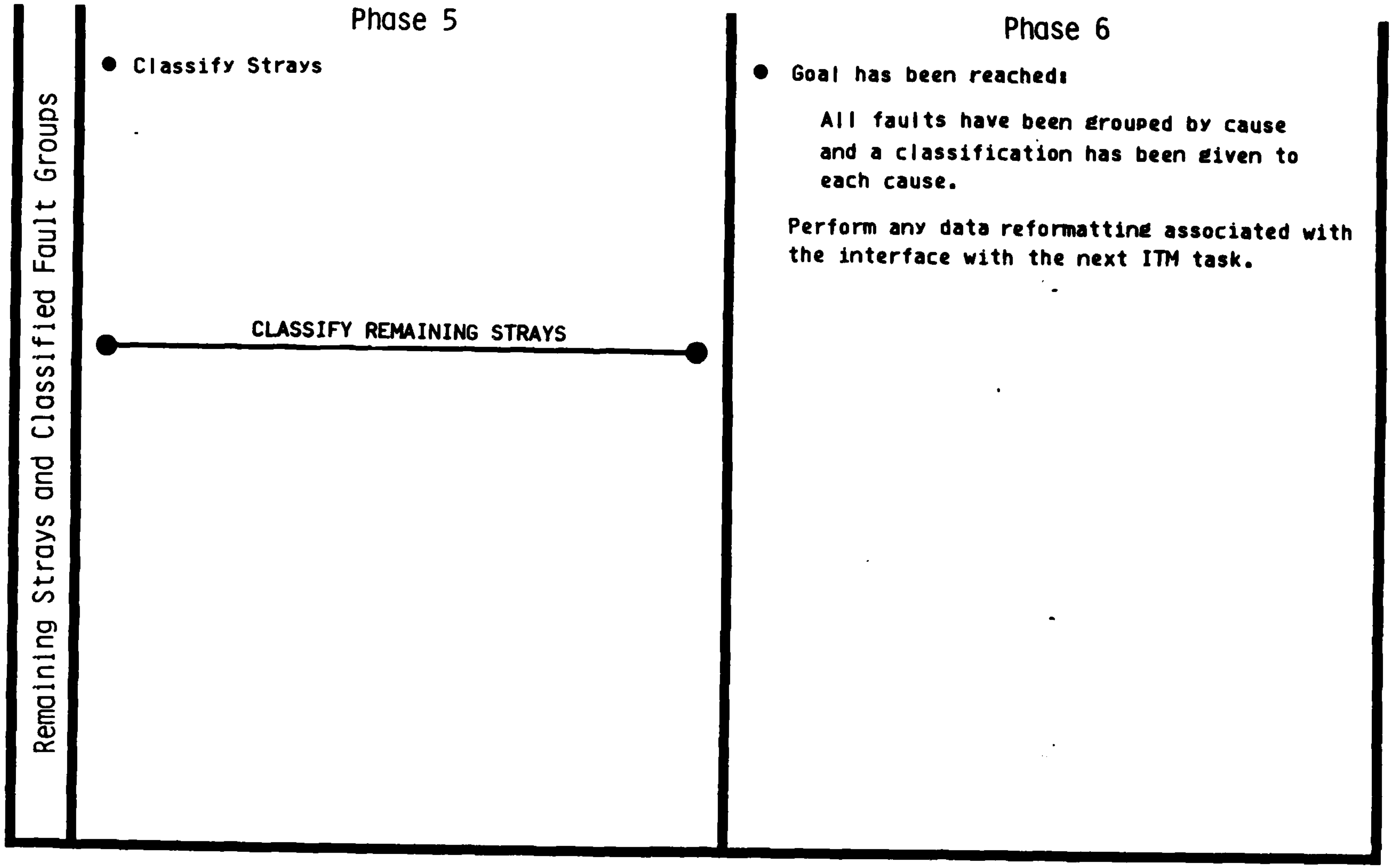
<u>TERMINOLOGY</u>
FAULT = TEST FAILURE
CAUSE = THE CAUSE OF SEVERAL TEST FAILURES

Figure 39. Phases of the Classification Expert System


Phase 3
● Make a preliminary grouping of the faults into categories representing effects of the same basic failure
Fault Data
IT WAS THE SAME TEST FAILING EACH TIME
Group All Together As One Cause
IT WAS A DIFFERENT TEST FAILING EACH TIME
Group According To Time Dependencies
THE SAME TEST FAILED SEVERAL TIMES AND SOME OTHER TESTS FAILED
Group According To Time Dependencies And Repeated Test Failures
Fault Groups and Strays
Phase 4
● For each group, classify as hard intermittent, etc.
● Use inferences to add strays to groups
Groups
CLASSIFY GROUP AS HARD
CLASSIFY GROUP AS DEGRADATION
CLASSIFY GROUP AS TRANSIENT
CLASSIFY GROUP AS INTERMITTENT
CLASSIFY GROUP AS FALSE ALARM #1
CLASSIFY GROUP AS FALSE ALARM #2
Strays
ADD STRAYS TO GROUPS
Remaining Strays and Classified Fault Groups

NOTE: These two tasks may not be decomposable. The addition of a stray to a group might depend on the classification of the group. Likewise, the classification of a group may change if it is associated with another (stray) fault.

Figure 39. (Continued)



Remaining Strays and Classified Fault Groups
Phase 5
CLASSIFY REMAINING STRAYS
Phase 6

- Classify Strays

- Goal has been reached:

  All faults have been grouped by cause and a classification has been given to each cause.

  Perform any data reformatting associated with the interface with the next ITM task.

Figure 39. (Continued)

or will be a "stray" (that is, there is no evidence that the cause of the fault is common with any other fault). The program will require interaction with the mission-to-mission fault data catalog during phases 3, 4, and 5.

d. Classification and final grouping. Classification and grouping actions are both active during this phase. Rules governing classification will allow each group of test failures to be labeled as hard, degraded, transient, intermittent, or false alarm. This labeling requires (1) correlation of fault occurrences with environmental and operational factors, (2) examination of the fault histories, and (3) specialized expert rules for the IFF subsystem. In parallel with this task, grouping rules may act to add stray faults to established groups. These rules would use additional data derived from the classification of the groups.

e. Classification of remaining strays. Here, any faults which have not been associated with other faults will be assigned to their own, individual groups, and classified. Most of these will be transients, intermittents, or false alarms, but fault histories will be checked to determine the seriousness of the failure.

f. Interface with fault isolation. Here, all output data will be reformatted for use in the later phases of POTP.

The problem representation has also been completed from the standpoint of states, moves and goals. This approach to defining an AI system is documented in Principles of Artificial Intelligence (11). For the classification type of expert system, the states are all the intermediate levels of understanding between initialization and final classification. A state, for instance, can represent a given preliminary grouping, how many of the groups have been classified, or the fact that a stray fault should be added to a certain group. The moves in the expert system are rules that contain the grouping and classification knowledge. A typical rule could be: if a fault is occurring discontinuously, but more frequently over time, then it should be classified as intermittent. The goal of the expert system is to have each fault assigned to a group (either by itself or with other faults) and to have each group assigned a classification defining the nature of the faults in the group. Additionally, these goal conditions must be met with sufficient level of certainty. A typical goal might be faults 1, 3, and 5 result from a hard fault, fault 2 is a false alarm (category 1), and fault 4 is a transient fault.

Knowledge representation in the expert system will tend to be mostly empirical. Causal knowledge usually requires a great deal more resources, so it will be incorporated sparingly. One place where it will be used is in the representation of the system architecture—the system will know what the LRUs are and how they are interconnected. This will allow more general rules, such as: if a power supply is showing intermittent failures, then the LRU's it powers are likely to exhibit failures and these failures should be grouped together. Note that this rule is not fully causal; it makes no attempt to explain why the two types of faults are linked.

8.6.3 Rule Development

Many of the rules necessary for the failure classification expert system have been defined. Rule development is an iterative process throughout the implementation of the expert system. The rules included here should be used to start development of the fault classification expert system. They will need to be modified and new rules added until they form a logical network that can support the expert functions. The initial implementation will handle the majority of situations. More rules will be added as testing continues and new problems are encountered.

The sources of the rules are the E-3 Technical Orders and interviews with E-3 IFF system experts. Some rules are subsystem independent and some are subsystem dependent. The subsystem independent rules apply to all subsystems and the subsystem dependent rules apply to a specific subsystem, in this case the E-3 IFF. Also, rules will either be grouping rules or classification rules. The rules are formatted as if-then statements but may be reformatted as implementation progresses. The subsystem independent rules are—

Grouping rules:

a. If failures occur in the same time frame, then they probably belong to the same group (possibly belong in separate groups).

b. If failures of the same set of tests (or individual tests) occur repeatedly, then they probably belong in the same group (possibly belong in separate group).

c. If two or more failures occur in the same time frame and they are associated with the same unit, then they probably belong in the same group (possibly belong in separate groups).

d. If a failure occurs in a unit that receives an input from another unit that is indicating a failure, then the two failures may belong to the same group.

e. If x number of failure indications occur at one time frame and a subset of x occur at some other time, then all of the x failure indications are probably in the same group.

Classification rules:

a. If a failure occurs only once, then it is a transient fault.

b. If a failure occurs more than once but the equipment functions properly between failures and does not fail at all during the time the it is in one or more specific modes of operation, then the fault is intermittent.

c. If a failure occurs more than once and the equipment functions between failures and fails all the time it is in one or more specific modes of operation, then it is probably a hard fault (possibly an intermittent fault).

d. If a failure occurs all the time, then it is a hard fault.

e. If a failure occurs intermittently and is a known false alarming test, then it is probably a false alarm (type 1) (possibly an intermittent fault).

f. If a failure occurs one or more times and the temperature of the devices is higher or lower than normal and the temperature is within specification, then the failure is probably an intermittent failure (possibly a false alarm (type 2)).

g. If a failure occurs one or more times and the temperature of the device is higher or lower than normal and the temperature is outside of specification, then the failure is probably a false alarm (type 2) (possibly an intermittent fault).

h. If a failure of a test-to-limits-type test is out of the normal range, but well within the specified range, then it is a degradation (type 1).

i. If a failure of a test-to-limits-type test is out of the specified range by a small amount and otherwise operates correctly, then it is a degradation (type 2).

j. If a failure of a test-to-limits-type test is out of the specified range by a small amount and other failures are indicated then it is probably a hard fault (possibly a degradation (type 2)).

The following are the subsystem dependent rules for the E-3 IFF. They are listed by component with additional background information.

IFF receiver-transmitter rules

The R/T contains its own low-voltage power supply powered by aircraft power. Seven of the R/T's 14 tests monitor the seven output voltages from the power supply. These tests check voltage level only, not ripple.

Derived rules:

a. If one or more of the R/T power supply tests (TP 12, 13, 15, 16, 17, 18 or 19) fails, but no radar target data processor test fails and no other R/T test fails, then the R/T is degraded, out-of-specification (classification rule).

b. If one or more of the R/T power supply tests (TP 12, 13, 15, 16, 17, 18 or 19) fails and one or more other R/T tests fails, then a faulty power supply is probably the cause (grouping rule).

IFF power supply rules

Each has five outputs:

+5V
+12V
-12V
+25V
+60V

All are provided to the radar target data processor. In addition the +25V powers the rf switch. Each power supply has 10 tests, two for each voltage, one on each side of the isolation diode. These tests check voltage level only, not ripple.

Derived rules:

a. If the 25V output fails, either the radar target data processor or the rf switch may be affected (grouping rule).

b. If the +5V, +12V, -12V, or +60V output fails, only the radar target data processor could be affected (grouping rule).

c. If one or more of the power supply tests fail, but no radar target data processor test fails, then the power supply is degraded, out-of-specification (classification rule).

d. If a diode output test (TP 4, 5, 6, 8, or 10) fails in one power supply, then it will fail in the other power supply (grouping rule).

e. If a diode input test (TP 1, 2, 3, 7, or 9) fails and the diode output test does not fail, then the radar target data processor should not fail (grouping rule).

f. If a diode output test (TP 4, 5, 6, 8, or 10) and a radar target data processor test both fail, then the radar target data processor faul: was probably caused by the power supply fault (grouping rule).

g. If a diode output test (TP 4, 5, 6, 8, or 10) fails in one power supply, and the corresponding test in the other power supply does not, then the BIT circuitry for the failed test has failed (grouping rule).

Target Data Processor Rules

The following testpoints (or groups of testpoints) are known to be potential false alarms:

208
213
101 and 301
409
102
101 and 104
101 and 103
101 and 105
101, 104, and 105
108
103
104
105
101 and 102
106
101, 103, and 105
101
101, 103, 104, and 105
329
330
329 and 330

Derived rules:

1. If any of the above tests or combinations of tests fails then it may be a false alarm (classification rule).

2. If any test 101 through 108, or any combination in the above list fails, and the test target received by the mission computer is correct, then the failure is a false alarm (classification rule).

As development progresses, the certainty factors will be incorporated into the rules. These are numbers between 0 and 1 to replace terms in the rules such as probably and possibly. These will need to be adjusted during testing.

8.6.4 Control Structure Development

The functions of classification and grouping are different in nature and the system needs to accommodate two different problem solving control structures. The grouping problem is basically one of synthesis; the system must form the most logical groups from the inflight fault records. The classification problem is basically analysis; the system must deduce the root cause of several fault symptoms. The general concensus of expert systems builders is that synthesis is best implemented by forward chaining and that analysis is best performed by backward chaining. This general rule of thumb.suggests a combined forward and backward chaining control structure with a fault-grouping subgoal as the central connection point of the two chains. That is, the system will work forward (from the fault data) until groups have been formed and then will work backward (from hypothesized fault causes) to determine the nature of the fault group.

. Decisions were made regarding control structure search strategies. These strategies define how the rules will be organized, when a rule or set of rules will be applied, and (to some degree) the structure and content of the rules. A brief description of decisions made and their impact follows:

a. The system will perform a hierarchical, rather than an abstract, search. This means that at any point in the process there will be only a limited set of rules that should be tested for applicability. Both the classification and grouping problems are structured enough that it is not necessary to check any of the more primitive and basic rules when at more advanced levels of reasoning.

b. There will be multiple lines of reasoning to diagnose system faults. This requirement matches the nature of avionic expert knowledge. That is, the expert has general rules of thumb that he can use to solve most problems, but in many situations, he will use more specialized knowledge to solve the problem more efficiently. Therefore, the rules may provide multiple paths for arriving at the same conclusion. For example, the general grouping algorithm holds that test failures occurring at the same time are probably the result of the same cause, and that all repeated test failures for the same flight have the same cause. However, the expert has additional rules for grouping power supply faults with processor faults and for recognizing common false alarm groups.

c. The system will use "fuzzy" models rather than deterministic. These methods must be used in grouping to allow alternate hypothesis of groups. They will be used in the classification section to implement fuzzy rules such as "The hotter an LRU is, the more likely it is to fail."

d. The expert system will not need to have a backtracking capability. Instead, it will have an irreversible rule application scheme. The grouping rules, for instance, will generate and evaluate alternative groupings in parallel, so that all logical groups will be tabulated at the time of passage through a grouping node.

e. There will be heuristic control over some actions of the system. Once into phase 4 (see fig. 39) of processing, there will need to be control over which of the two major functions is to act. That is, once a grouping has begun, it should not be interrupted by a classification action (and vice versa). The heuristics will determine which of the two actions should be initiated, as well as prevent interruptions once an action has been initiated.

Requirements have been developed for the classifi-ation and grouping heuristics. These requirements implement the problem representation shown in figure 39. Ground rules of the scheme are—

a. Faults are assigned to groups, then groups given classifications.

b. After preliminary grouping, both the classification and grouping actions are active. Grouping rules that are performed afterwards differ in that they depend on output of the classification actions (e.g., hard power supply test failures cause intermittent target data processor failures).

c. The system must be capable of choosing between alternative groupings of faults. This should be accomplished by maintaining a fault in multiple groups simultaneously and having certainty factors for each group, or by ruling out all but one alternative before the fault is grouped. The ITM system will use the former method because it allows for grouping in the absence of a clear-cut choice of alternatives.

Fault grouping approach

The grouping approach varies with the fault conditions. The various approaches for grouping two faults together are—

Case 1: Neither fault is bound to a group:

a. Create a new group.
b. Assign each fault to the new group.

Case 2: One fault is bound to a group and one is not:

a. Determine if the unbound fault logically fits into the other fault's group. (i.e., is the suspect cause of the new group the same as the suspect cause of the existing grouping). If so, add the new fault to the existing group. If not, create an independent group and assign the two new faults to the group. Modify the certainty factors for the two groups.

b. If the bound fault is bound to more than one group, the new fault should be tested for addition to each group before any new groups are created.

Case 3: Both faults are already bound to groups:

a. If the two groups have the same cause, merge them into a single group.

b. If not, form a third group containing the two faults. Modify the certainty factors of the three groups.

Fault classification approach

Approaches for classifying faults are—

Case 1: The fault is not assigned to a group already:

a. Create a new group.

b. Assign the fault to the group.

c. Assign a classification to the group.

Case 2: The fault is assigned to a group already:

a. If there are no other faults in the group, assign the classification to the group.

b. If there are other faults, determine if the classification is consistent with the other faults.

If so, classify the group.

If not, the classification cannot be assigned to this group.

c. Repeat step b for all groups that the fault is assigned. If the classification does not fit any group, create a new group as in case 1, then modify the certainty factors of all the groups containing the fault.

8.6.5 Sizing and Timing

Sizing and timing estimates are necessary in development of architectures when resource allocation is a critical design factor. Avionic architectures are one area where tradeoffs are necessary to determine which functions should fit into the limited resources. Research in the expert systems field, to date, has not explored methods of estimation of resources because most of the developed programs run in environments where resources are not critical.

The problem with sizing and timing stems from the fact that the AI languages such as LISP are not structured in the same manner as conventional languages such as JOVIAL or ADA. The primary difference is that data has no predefined address in memory. That is, a memory address is allocated to a variable when it is assigned a value by the program's instructions during a run. A variable is deleted from memory when it is no longer active in the program; if it becomes active later, it is allocated to a new memory location. Thus, the working memory of the program is continually growing and shrinking as the program is executing. Another difference is that there is not structured flow through the statements (rules) of an expert system; the execution of a rule depends primarily on whether or not its 'if' conditions have been met and if it satisfies precedence rules for breaking ties when more than one rule applies. While this characteristic aids easy development and modularity of the expert system, it makes it very hard to determine the number of rule applications necessary for problem solution. Also, it is not apparent how many rules will have satisfied 'if' conditions at any one time; so there is no way to accurately estimate the time required to determine which rule will be applied. These complications make it very difficult to predict the timing and sizing before the expert system has been developed.

Given the difficult task of estimation under these conditions, a first-order approximation of timing and sizing was attempted using the following approach: (1) a conventional analytical model for the two parameters was developed; (2) the factors used in the model were studied to determine what characteristics of the expert system had an impact on them; (3) where possible, the factors were estimated directly from knowledge of the ITM expert system and the nature of the factors; (4) if direct estimates were not possible, comparisons were drawn with the OPS5 expert systems environment and with the R1 expert system(15) because they are somewhat similar to the ITM application, and because statistics for these systems are available. In some of the following estimations, data was extracted from systems using LISP processors rather than conventional 16-bit aerospace processors; this may tend to make estimates somewhat optimistic. The processors considered as a baseline for this effort are MIL-STD-1750A processors with 256K memory operating at 1 MIP.

The conventional analytical approach to timing and sizing would be to model timing as:

(program execution time) = (rule execution time)
x (number of rule applications for problem solution)

And to model sizing as:

(program size) = (size of control kernel) + (size of rule base)
+ (worst case size of working memory)

Each of the right-hand-side factors is discussed below:

Rule execution time.
This factor can be broken down into three components:

a. The time required to determine which rules have satisfied left-hand sides. This is proportional to the number of rules in the system.

b. The time required to determine which of the "matched" rules will be applied during the current cycle. This action, called conflict resolution, takes an amount of time proportional to the number of rules matched and is thus related to some degree to the number of rules in the system. However, a more structured or hierarchic system will match fewer rules per cycle and thus be faster at conflict resolution.

c. The time required to execute the right-hand side of the selected rule. This parameter is fairly constant between expert systems and is insignificant when compared to the first two parameters.

As a rough comparison, the R1 expert system, which has between 30 and 256 rules active at any one time, consumes about 0.14 seconds per rule application cycle, running on a PDP-10.

Number of rule applications for problem resolution—would vary from run to run and it would depend on the number of fault occurrences during flight, the number of fault causes or groups, the ambiguity of the fault indicators, the uniqueness of the rules in the system, and probably some other factors which are not yet apparent. As a rough estimate, one could look at statistics gathered from the R1 expert system (15). This system, which has 776 rules, uses about 41% of its rules for a given run and has about 1.4 times as many rule applications as there are rules (i.e., there are repeated applications of some rules). If the ITM expert system had similar characteristics and contained the estimated 500 rules, then the estimate would be 500 x 1.4 = 700 rule applications to complete processing.

Using the data above and making the assumption that the two factors will be similar to those of the ITM system, the timing estimate would be—

$$\begin{aligned}\text{total time} &= (0.14 \text{ sec/rule application}) \times (700 \text{ rule applications}) \\ &= 98 \text{ sec}\end{aligned}$$

Timing, therefore, does not appear to be a problem in a postflight environment.

Size of control kernel—during the development of the ITM expert system, the OPS5 production system development tool was used as a guideline. About one third of this tool is used to implement the control structure (recognize-act cycle) of an expert system. The size of the OPS5 load module on a VAX 11/780 is 580,000 bytes. A rough estimate of the size of a control kernel necessary for a deliverable system would be—

$$(1/3)\ (580\text{K}) = 193.3\text{K bytes} = 96.7\text{K 16-bit words}$$

This is about 38% of the 256K mission processor and about 159K words of memory would remain for the rules and for working memory.

Size of the rule base—based on the estimate that each rule requires 100 words of memory for storage and that there are 500 rules in the system, the size of the rule base would be 50K words.

Worst case size of working memory—the size of the working memory depends on the number of fault indications that occurred during the flight, the real number of groups of faults, and the apparent number of possible groups and classifications that the system detects.

Using the above data, the sizing estimate would be:

96.7K words for the control kernel
+50K words for the rule base
+109.3K words for working mass memory (maximum available space)
256K -- or less to fit in avionic processor

The sizing estimate above leaves only 109.3K of memory left for working data items. Given the speculative nature of the other sizing factor estimates, there is a high risk factor associated with developing this system in a memory-constrained environment. There are, however, several approaches to relieving the sizing risk. These are discussed in the following section. These approaches are only necessary if the expert system is to be constrained to 256K processors, such as those currently being proposed for the PAVE PILLAR program.

8.6.6 Software Structure

The allocation of resources to the expert system is difficult because of the inability to determine the size of the software involved. Of course, the ideal situation would be to run the expert system program with the rest of the OTP. But, given the size of current expert systems, this would be impractical. Several alternatives are possible for implementation of the expert system. The alternatives provide for location of the expert system in the avionics and compensate for risks of overflowing allocated resources.

Location of the expert system

a. The expert system could run in the main mission processor as a separate load module from the OTP. Under this scheme, the system is loaded during postflight and collects its input data from the SMM. It is able to utilize all of the processor's resources (256K, 1 MIPS). It stores its output on SMM after it finished its task, and then the remaining part of the postflight program is loaded into the mission processor.

b. The expert system could run in the backup mission processor. This scheme allows the OTP to remain in the primary processor, and the time necessary to perform the expert system overlay to be saved. There is a drawback to this scheme. If one of the mission processors failed, then there would be no place to perform the classification function without reverting to scheme a.

c. A third approach is to incorporate an AI processor into the avionic architecture. Although this is a high-cost alternative, there would be large gains in the speed of the expert system and new possibilities would open for the use of the AI processor during flight. For instance, it could be used for ITM tasks such as intelligent fault monitoring or for other tasks, such as pilot workload management.

Memory utilization risks

a. Constraining the expert system to one processor is the simplest solution to the problem of an unpredictable working-memory size. This requires the system to have rules that delete some working-memory elements (such as groups with low certainty factors) when all memory was filled. Such a system still runs the risk of destroying useful data.

b. Allowing processor memory overflow to be stored on the SMM keeps potentially useful data from being destroyed. This system has rules that allowed portions of the working memory to be stored on the SMM unit. The working-memory elements could be accessed in the standard manner, or the process might be optimized by shifting blocks of data into and out of the mass memory. The latter process requires some development before it could be integrated with current expert system control logics.

c. A dual processor approach also provides more available working memory area. The scheme not only stores data elements in the backup computer, but might also be able to perform some rule matching. Application of this approach requires further research into decomposition strategies and communication links necessary to implement the expert system in a parallel processing manner.

8.7 RECOMMENDATIONS FOR FURTHER AI RESEARCH

The following are recommendations for future research funding in artificial intelligence. They are meant to plug some specific holes in AI technology which became apparent during the development of the ITM expert system.

a. Fully develop a representative expert system with the developer closely monitoring the figures of merit that are needed for military deliverable software. That is, unlike academic and some commercial developers, military developers should be keenly interested in sizing, timing, reliability, maintainability, development time, etc.

b. Conduct a study that analyzes and evaluates techniques for integration of AI programs into fighter avionics. Specifically, is it possible to efficiently use conventional processors for AI tasks or would an avionic AI processor be required? If a conventional processor is used, how will software be developed for it? Should there be a standard LISP compiler for the military processor, or could a conventional language such as ADA or JOVIAL be used?

c. Evaluate the feasibility of an inflight fault classification expert system. This would involve addressing not only the much more restrictive resource and timing problems but also the continual data acquisition and update feature.

d. During the AI investigation, the subject of self-improving diagnostics was examined. This is a feature of an expert system that could change the rules based on new results it receives from interaction with the maintenance technician on problems the machine cannot diagnose. The subject of self-improving diagnostics was not pursued, since ITM was concerned with onboard test capability; and it is inappropriate to permit each fighter to learn new diagnostics based on its own limited sample of data. Self-improving diagnostics should, however, be explored further for application at a central maintenance facility to improve fleet-wide diagnostics based on fleet-wide test experience.

9.0 CONCLUSIONS

The Integrated Testing and Maintenance Technologies study effort defined requirements for the onboard test and maintenance system for the 1990's tactical fighter. These requirements are documented in the ITM system specification.

This report provides the background and analyses that led to the ITM system requirements. Problems with test and maintenance systems were analyzed and current tactical fighters were evaluated to determine where improvements could be made. The anticipated avionic architecture and mission for the 1990's tactical fighter were evaluated to determine new demands on the test and maintenance system. From these, the requirements for ITM were developed by starting at the highest level, defining the required capabilities, and specifying requirements down to the level of what tests to perform and how the system operates.

The system as specified is primarily a test data collection system, with tests provided by the subsystem and test data processing provided by the subsystem software. The improvements over current systems provided for in the specification include—

a. Better filtering of nuisance alarms from the pilot. This includes filtering of intermittent failure reports when they do not affect the mission and suppression of reports of noncritical failures during critical flight phases.

b. Better isolations of intermittent faults. This is accomplished by inflight recording and postflight processing of fault data and associated data, including equipment temperature, aircraft flight dynamics, equipment operating modes, and subsystem modes.

c. Incorporation of more extensive system-level tests. These include reasonableness tests of mission operational data, statistical tests on the innovation vector from a Kalman filter, and voter tests where applicable.

d. Variable test tolerances. For those tests that are measurements of performance against limits (e.g., power supply voltage, transmitter power) the limits are stored in the system software instead of in the subsystem hardware.

e. Interactive participation by the maintenance technician. Besides selecting the test sequences and observing results, the maintenance technician receives test data and the diagnostic decisions the system used in isolating a problem. These are used by the maintenance technician along with the capability to access additional test data to isolate the difficult problems. The interaction with the system provides "on the job training" for the maintenance technician instead of relegating him to the role of a "button pusher."

f. Offline use of test data. Test data that will help the depot to isolate failures or determine the cause of them is stored for access at the depot. This data can also be used to create a failure history to develop failure trends.

Another conclusion of the ITM effort is that artificial intelligence, expert systems, is potentially useful for application to testing and maintenance problems and appears practical for postflight analysis applications in future generation tactical fighter aircraft. The benefit of an expert system is that it can help solve problems normally requiring the experience of the system designer or an experienced maintenance technician.

While the use of expert systems appears beneficial and practical, it still requires more development work before it can be applied on a tactical fighter development contract for integration with its avionic system. No systems with AI have yet been developed for avionics, and most military applications are just now getting started. This makes specification and management of its development for a project difficult and risky. Continued development of the expert system described in this study or a similar development effort on an R&D basis is recommended as a risk reduction measure before insertion in a mainstream program.

REFERENCES

1. "DAIS Integrated Test System," GRR-005-1176, Lear Siegler, Inc./Instrument Division, Grand Rapids, Michigan, 1976.

2. "Design-for-Repair Concept Definition," AFAL-TR-1130, Hughes Aircraft Company, August, 1979.

3. "Built-in Test Equipment Requirements Workshop," Institute for Defense Analysis, Paper P-1600, August, 1981.

4. "Guide 3—Avionics Testability Design Guide," from Modular Automatic Test Equipment Guides, Volume 3—MATE Developed Tools, Final Draft, Sperry Corp., June, 1981.

5. "Mission Demonstration Specification for the Advanced System Avionics—Operational Readiness Mission," SSA00501001, June, 1981.

6. "Mission Demonstration Specification for the Advanced System Avionics—Survivable Penetration," SSA00501002, June, 1981.

7. "Mission Demonstration Specification for the Advanced System Avionics—Survivable Strike Mission," SSA00501003, June, 1981.

8. "On-Aircraft Testing Techniques Final Technical Report," Lear Siegler, 1980.

9. "General Requirements for Maintainability of Avionics Equipment and Systems," AR-10A, Naval Air Systems Command, March, 1976.

10. "Artificial Intelligence," Rich, E., McGraw-Hill Inc., 1983.

11. "Principles of Artificial Intelligence", Nilsson, N.J., Tioga Publishing, 1980.

12. "Handbook of Artificial Intelligence," Barr, A. and Feigenbaum, E., William Kaufmann Inc., 1982.

13. "Computer-Based Medical Consultations: MYCIN," Shortliffe, E.H., New York: American Elsevier, 1976.

14. "An Overview of Expert Systems," Gevarter, W., U.S. National Bureau of Standards, 1982.

15. "R1: A Rule-Based Configurer of Computer Systems," McDermott, J., Artificial Intelligence Journal, 1982.

16. "System Specification for the Digital Avionics Information System," SA 120 100, April, 1979.

17. "DAIS Technical Manual for Remote Terminals (15538)," MA 321 301, February, 1981.

18. "System Segment Specification for DAIS Control and Display Subsystem," SA 501 200, March, 1977.

19. "Prime Item Development Specification for AN/AYK-15A Digital Processor," Part 1, Type B1, SA 421 205, December, 1979.

20. "System Control Procedures (1553B, AN/AYKK-15)," MA 221 200, October, 1979.

21. "Top Down Built-in Test Architecture Study," LR 29523, Lockheed–California Company, April, 1980.

22. "BIT False Alarms: An Important Factor in Operational Readiness," John Malcolm, Proc. Annual Reliability and Maintainability Symposium, 1982.

23. "Modular Automatic Test Equipment Guides," Sperry Corp., June, 1981.

24. "Design-for-Repair Concept Definition," AFAL-TR-79-1130, Hughes Aircraft Company, August, 1979.

25. "Maintenance Techniques on the F-15 and F-4 Aircraft," Trip Report, Memo for Record, AFWAL/AAAS-2, September, 1981.

26. "IDT: An Intelligent Diagnostic Tool," Shubin, H. and Ulrich, J.W., AAAI-82 Proceedings.

27. "REACTOR: An Expert System for Diagnosis and Treatment of Nuclear Reactor Accidents," Nelson, W.R., AAAI-82 Proceedings

28. "LISP," Winston, P.H. and Horn, B.K., Addison-Wesley Publishing Company, 1981.

29. "A CSA Model-Based Nuclear Power Plant Consultant," Underwood, W.E., AAAI-82 Proceedings.

30. "Diagnosis Using Hierarchial Design Models," Genesereth, M.R., AAAI-82 Proceedings.

APPENDIX A

ANALYSIS OF DAIS

The Digital Avionics Information System (DAIS) was developed under direction of AFWAL as a laboratory testbed for 1980's-technology fighter avionic systems. The new features of DAIS included the implementation of a MIL-STD-1553B-oriented system and development and definition of standard elements.

A.1 DAIS System Architecture

The DAIS system architecture is shown in figure A1. The system comprises a 1553B-protocol multiplex system, one to four mission processors, a controls and displays subsystem, and a DEC-10 support facility that simulates the avionic sensor suite, the system mass memory (SMM), and the stores launcher processor. The various system elements and their BIT capabilities are discussed in the following sections.

A.1.1 Multiplex System

The DAIS multiplex system consists of the 1553B data bus, remote terminals, the bus active discrete module, and the processor address discrete module.

A.1.1.1 1553B Data Bus

Communication through the system is via MIL-STD-1553B bus protocol. The master processor controls all bus traffic between local processors, remote terminals, and SMM. The three types of message words—command, status, and data—are described in detail in the system specification for DAIS (16). Message formats are shown in figure A2 and word formats are shown in figure A3. Three important areas of the MIL-STD-1553B data bus affecting DTM design—status word, mode commands, and cycle timing—are further described in the following paragraphs.

Status words—any device other than the master controller will send a status word to the controller after it has received data or before it transmits them. No status word is sent if there is a message error when receiving data. Also, the master can command a remote device to send only its status word.

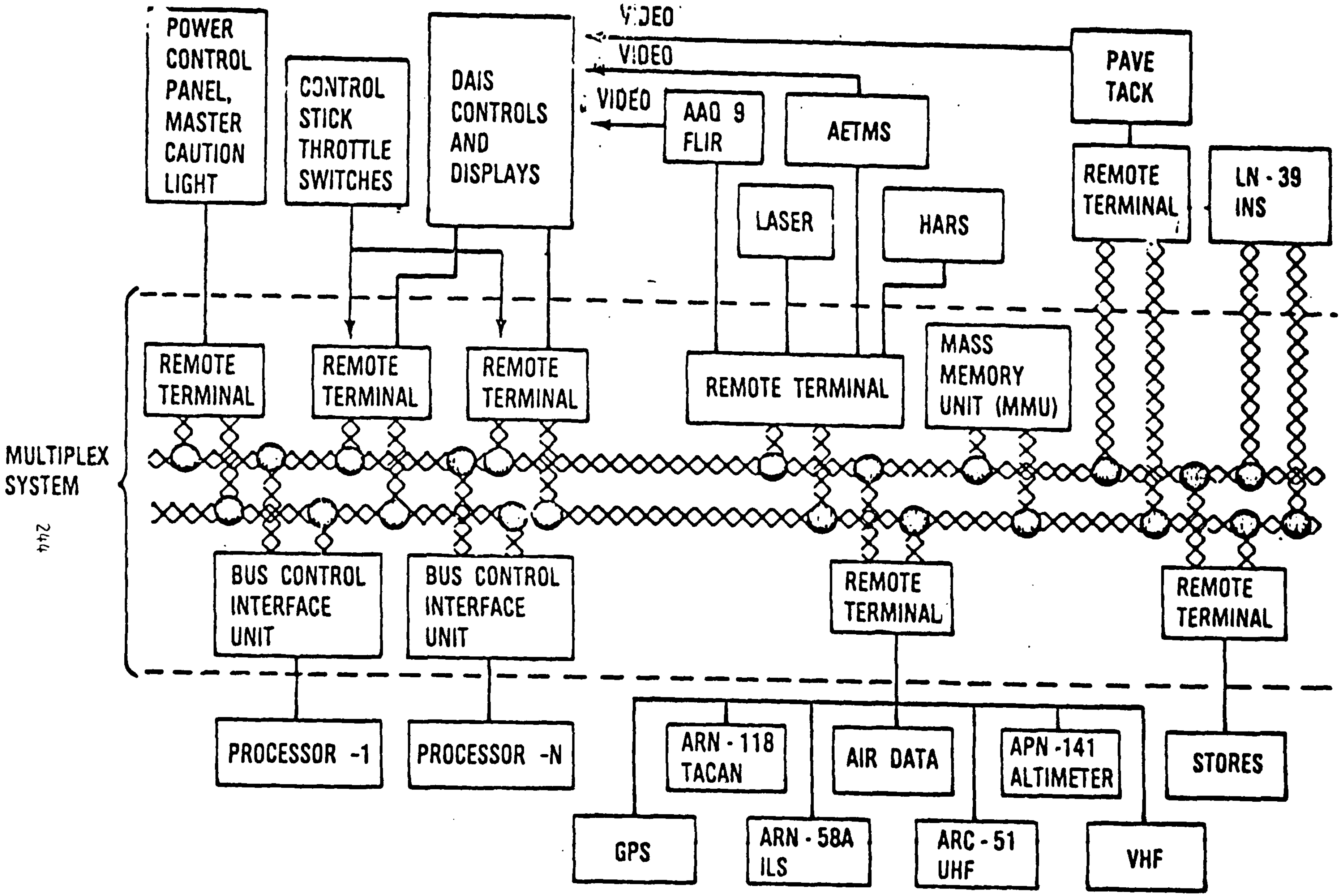
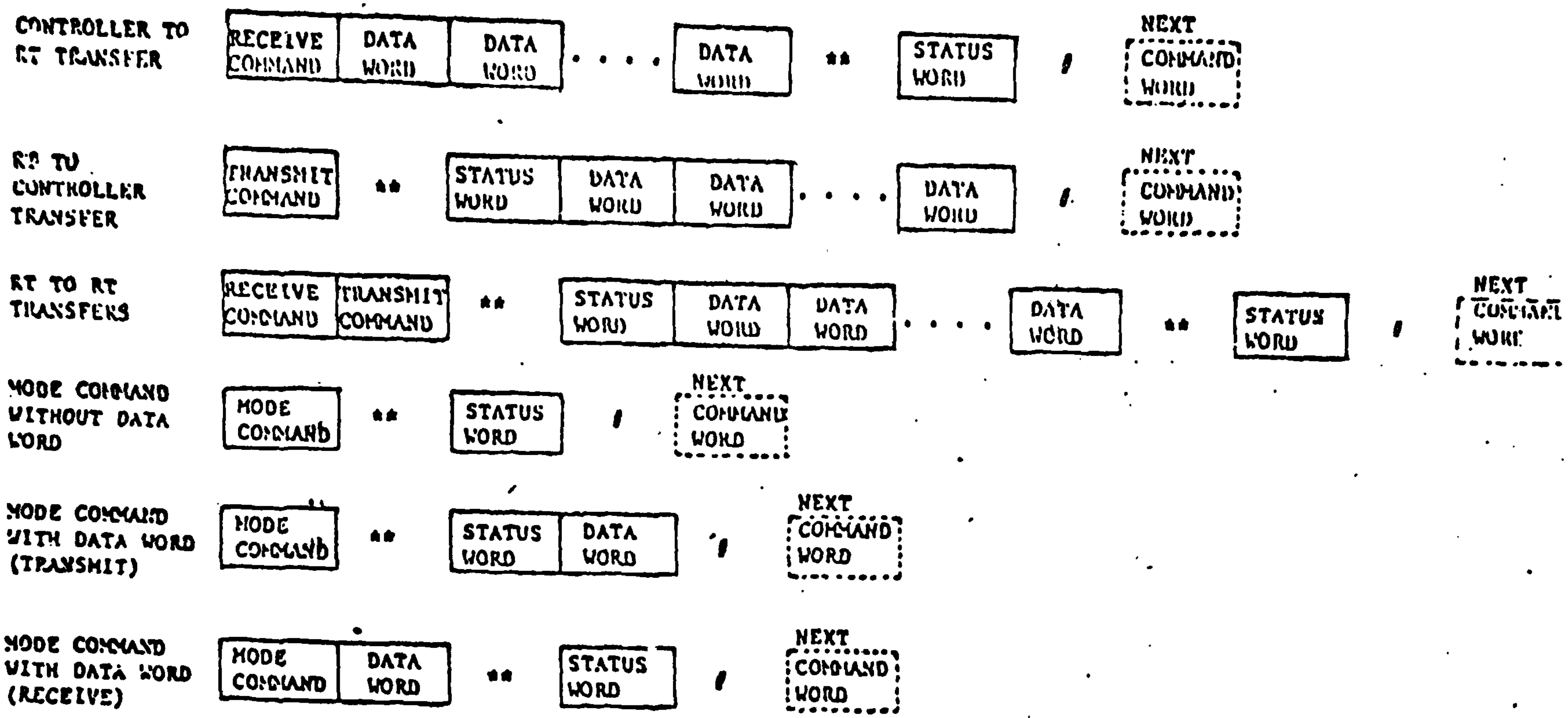
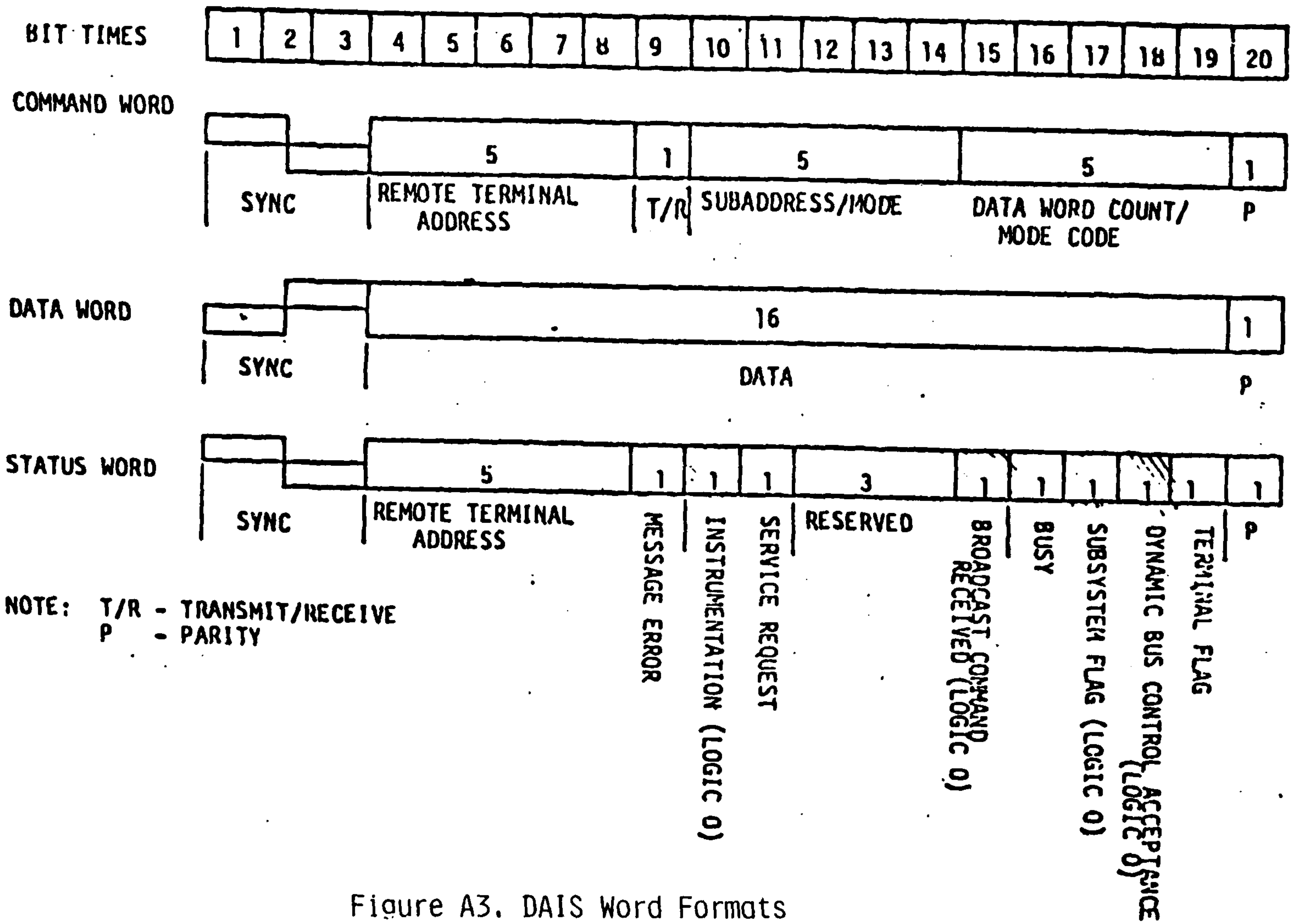


Figure A1. DAIS System Architecture



NOTE: # INTERMESSAGE GAP
** RESPONSE TIME

Figure A2. DAIS Message Formats



NOTE: T/R - TRANSMIT/RECEIVE
P - PARITY

Figure A3. DAIS Word Formats

The status word contains these ITM-related bits:

a. Message error—indicates the last received message did not pass validity tests.

b. Busy bit—is set when a remote processor can neither transmit nor receive data. It can be set by the remote controller unit or by the remote processor software, but can only be cleared by the software. It does not necessarily indicate a failure.

c. Terminal fail bit—this bit is set when the remote device has detected a self-test failure.

d. Service request bit—indicates activity at a remote device.

There are some MIL-STD-1553B options and related status word bits that have not been implemented in the DAIS system. These include:

a. Reserved bits—three bits have been reserved for future use.

b. Broadcast command received bit—the broadcast option is a message that is sent from a bus controller or a remote terminal to more than one remote terminal. It is not used in the DAIS system.

c. Subsystem flag—indicates a subsystem fault conditon and alerts the bus controller to potentially invalid data. In the case of multiple subsystems interfaced to a single remote terminal (RT), the individual subsystem flags will be ORed together to form this bit. The designer must make provisions in a separate data word to identify the specific reporting subsystem.

d. Dynamic bus control acceptance bit—indicates that the remote device has accepted dynamic control of the data bus.

Mode commands—commands used by the master controller to diagnose remote device status. A remote device will respond to a mode command by sending a status word and optional data words. Mode commands are shown in figure A4.

MODE OPERATION	VALID FOR	MULTIPLEX BUS PROTOCOL
Undefined (0,6,7,9-15,20-31)	see Table II	TX/RX COMMAND * STATUS[1]
Synchronize (1)	Remote Device	TX COMMAND * STATUS[5]
Transmit Status (2)	Remote Device	TX COMMAND * STATUS[6]
Initiate Self Test (3)	Remote Device	TX COMMAND * STATUS[3]
Transmitter Shutdown (4)	Remote Device	TX COMMAND * STATUS
Override Transmitter Shutdown (5)	Remote Device	TX COMMAND * STATUS
Reset Remote Terminal (8)	Remote Device	TX COMMAND * STATUS
Transmit Vector Word (16)	Remote Device	TX COMMAND * STATUS VECTOR WORD[4]
Synchronize (17)	Remote Device	RX COMMAND SYNCHRONIZE DATA * STATUS[7]
Transmit Last Command (18)	Remote Device	TX COMMAND * STATUS LAST COMMAND[2,6]
Transmit BIT WORD (19)	Remote Device	TX COMMAND * STATUS BIT WORD[6]

1. ME Bit SET IN STATUS
2. DO NOT UPDATE CURRENT LAST COMMAND
3. TF BIT SUPPRESSED FOR THIS STATUS

*Response time as specified

4. SR BIT SUPPRESSED AND DISABLED FOR THIS STATUS
5. SR BIT SUPPRESSED AND RE-ENABLED FOR THIS STATUS
6. DO NOT UPDATE OR CLEAR BIT REGISTER
7. SR BIT SUPPRESSED AND RE-ENABLED FOR RT ONLY

Figure A4. MUX-BUS Protocol For DAIS Mode Commands

These mode commands have been implemented on the DAIS baseline and can aid ITM functions:

a. Transmit status—the remote device will respond by sending its status word.

b. Initiate self test—will cause the remote device to reset its BIT word terminal failure field and begin self-test.

c. Transmit vector—causes the remote device to transmit its status word followed by a data word containing service request information.

d. Transmit last command—the remote device will respond by sending its status word and the last valid bus command received by that device.

e. Transmit BIT word—the remote device will send its status word and its BIT word. The message error field of the BIT word will not be cleared or updated when this mode command is sent.

The following MIL-STD-1553B mode commands have not been implemented for the DAIS system but may be useful for ITM.

a. Inhibit terminal fail bit and override inhibit terminal fail bit—control the remote device's ability to indicate a terminal failure.

b. Reserved commands—there are 17 mode commands that have been reserved for future use.

Major and minor cycle timing—The DAIS major cycle is 1 sec in duration and is divided into 128 minor cycles. Snychronous tasks are assigned to minor cycles by an offset from the first minor cycle and a period. The period must be a power of 2 (i.e., p = 1, 2, 4, 8, 16.....128).

a. Synchronization—master processor controls the synchronization of all processors in the system. A synchronize mode command is sent to each remote processor after both the minor cycle clock has expired and the synchronous bus list is complete; when the actual minor cycle clock has expired and the synchronous bus list is complete; and, finally, when the actual minor cycle number is updated. The master processor also keeps a theoretical minor cycle number, which is updated whenever the minor cycle clock expires. If, due to the synchronous bus list overrunning minor cycle frames, the actual minor cycle lags the theoretical minor cycle by more than a predefined number, the master processor flags itself as failed and performs self-test operations.

At the remote processor level, synchronization activities begin when the "synchronize" mode command causes an interrupt. If the minor cycle number sent from the master is the expected one, registers and pointer tables are updated in a normal fashion.

If the new minor cycle is not the expected one, the remote processor tabulates the error and sets the registers and pointer tables for the minor cycle received.

b. Synchronous input/output (I/O)—synchronous I/O in the DAIS system is handled by the bus control interface of each device and is driven by synchronous I/O tables indexed by minor cycle number.

c. Asynchronous I/O—multiplex units are capable of asynchronous I/O. Only the master processor can initiate asynchronous transmission. A remote device must have its service request bit set. The bit is recognized by the master after the next status word transition.

A.1.1.2 Remote Terminal

The RT performs testing and maintenance functions in three areas: bus data checks, self-tests, and subsystem interface data checks. Most errors are reported in the RT's Built-in Test Register (BITR). An item diagram for the RT is shown in figure A5.

For received data, the multiplex terminal units (MTU) check for valid syncronous signals, correct Manchester code and transition times, and for correct parity. A correctly received word is sent to the terminal control unit (TCU) via a parallel (16 bits + parity) holding register.

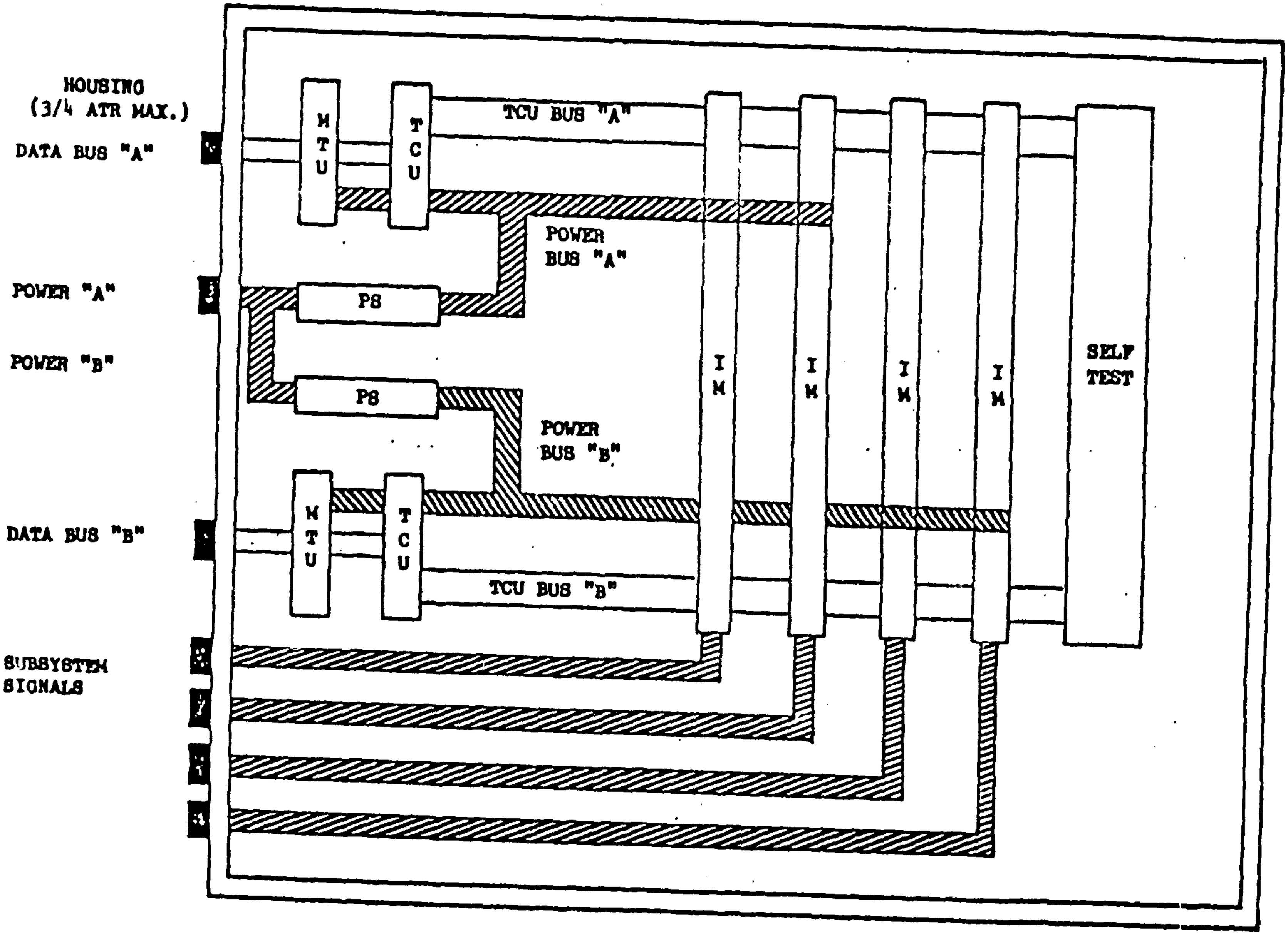


Figure A5. Remote Terminal Block Diagram

The TCU commands level error checking on the incoming data; errors detected are no data received, word count high, word count low, and invalid command.

For transmit operations, testing is through wraparound tests in the MTU. Data output to the bus is channeled back through the MTU receive logic validity checks. If any tests are failed—bit count high, bit count low, bad parity, bit validity—the TCU will set the message error bit in the status word. The word is sent after data transmission is complete. Additionally, if an MTU transmits continuously for more than 680 sec, it is disabled and the word is declared invalid.

Registers in the RT, which are important for ITM functions, are the BITR, the activity register, and the last command register. A description of the BITR can be found in the DAIS Technical Manual for Remote Terminals.(17) Activity register bits are programmed by the user to indicate successful data transfer from serial digital IM channels; they are reset after the corresponding channel data is read or by the reset remote terminal mode command. The last command register contains the last valid command received by the RT.

Each IM input channel is tested for validity. Most cards have a redundant test logic on the card; data are compared for validity before being passed to the TCU. Other input channels are tested at the TCU level (e.g. for synchro input, the TCU checks the identity $K^2\sin^2\theta + K^2\cos^2\theta = K^2$ to determine if the module is operating correctly).

For output IM's the subsystem interface data are returned to the TCU and compared with intended data.

A.1.2 Sensor Group

The DAIS laboratory implementation of avionic sensors is via software models in the DEC 10 support equipment. A summary of the testability and maintenance features of the avionics subsystems is in table A1. The remote terminal bus transmission rate is also given in that table.

TABLE A1. SENSOR TESTABILITY AND MAINTENANCE COMPARISONS

Operational Readiness mission					
Specified sensor	DAIS lab implementation	Function	Inputs* from RT	Outputs* to RT	ITM capability utilization
PAVE TACK	Generic laser ranger (software model)	Give slant range to objects 0 to 750,000 ft	Beam pointing and stabilization commands from pilot, range request	Slant range, range invalid signal	15 bits unused Output signal No BIT data
LN-39 INS	SKN 2416 INS (software model)	Provides processed inertial navigation data Can default to HARS function if INS computer fails	Asynchronous: Filter mode, NAV state vector, gyro bias corrections IR = 32: Vehicle dynamics data from other subsystems with validity bits	IR=32: Velocity platform attitude, heading, acceleration, direction cosines, longitude, altitude	Good fault detection Many spare output bits Input contains data that could be used for local fault monitoring and self-test
Air data sensors	Generic air data sensors (software model)	Provides temperature, pressure, angle-of-attack data to other systems	None	IR = 8: Total temperature Total pressure Static pressure Angle of attack	No BIT data
ARN-58A instrument landing set	ARN-58A (software model)	Aid pilot by tracking runway approach position	IR = 8: Channel select Power on	IR = 8: Glide slope deviation Localizer deviation Glide slope reliability Localizer reliability	No BIT Data No Spare bits

* These represent the sensor itself and not necessarily the software implementation. Iteration rate (IR) is the number of transmissions per second.

TABLE A1. SENSOR TESTABILITY AND MAINTENANCE COMPARISONS (CONTINUED)

Operational Readiness mission					
Specified sensor	DAIS lab implementation	Function	Inputs* from RT	Outputs* to RT	ITM capability utilization
ARN-118 TACAN tactical air navigation	ARN 118 (software model)	Provides bearing and slant range relative to TACAN stations (ground or air)	IR = 8: Channel select, initiate self test, mode control, power on channel	IR = 8: Range, bearing, status, parity bit, range rate	Self-test provides a complete test of TACAN (except antennas), 2 BIT status words for range, bearing, range rate, mode control
APN-141 radar altimeter	APN-141 (software model)	Continuous, highly accurate altitude for 0 to 5,000 ft	IR = 8: Low-altitude limit Low-altitude test signal Power Radar altitude test	IR = 8: Linear altitude Low-altitude warning Modified reliability	Range measures phi after successful self-test No spare bits.
ARC-51 UHF radio	ARC-51 model	AM voice communications or automatic direction finding	IR = 2: Frequency select Guard on Squelch disable Mode T/R Mode ADF	None	None

* These represent the sensor itself and not necessarily the software implementation. Iteration rate (IR) is the number of transmissions per second.

TABLE A1. SENSOR TESTABILITY AND MAINTENANCE COMPARISONS (CONTINUED)

Operational Readiness mission					
Specified sensor	DAIS lab implementation	Function	Inputs* from RT	Outputs* to RT	ITM capability utilization
	Aircraft status system	Provide pilot aircraft status data to pilot	None	IR = 8: Flap position Speed break position Main fuel tank quantity Transfer fuel tank quantity Fuel flow Each wheel position Weight off gear Fire warning	None Some spare bits
Heading and attitude reference system	SKN 2416 INS (software model in degraded mode)	Provide heading and attitude reference	See LN-39 INS description block	Roll, pitch, and azimuth attitudes	See LN-39 INS description block
AAQ-9 FLIR AETMS GPS VHF radio	No lab implementation				

* These represent the sensor itself and not necessarily the software implementation. Iteration rate (IR) is the number of transmissions per second.

A.1.3 Controls and Displays

The control and display (C&D) subsystem consists of a remote terminal used to interface a processor control panel (PCP) and a dual redundant pair of RT's for the display generation equipment, the displays, and the controls. A complete description of the LRU's and their functions is included in the system segment specification for DAIS control and display subsystem.(18)

A detailed list of the C&D equipment and its ITM-related I/O is given in table A2. The display devices each have "temperature high" and "display status" bits along with one to four spare status bits. Control devices have one power status bit each. The MPDG's are the only display support equipment that interface directly to the remote terminal, and they do all status reporting to an LRU level for that group. MPDG-1 and MPDG-2 have identifical I/O formats except they use different cards and channels. The ITM data for the MPDG's is in table A3. There are ample spare data bits on the IM cards.

The PCP-ACP-MCL panel and the PDP panel interface the system through a separate RT. These panels are closely related to testability and maintenance functions.

The AAQ-9 FLIR, AETMS, and PAVE TACK send video signals directly to the display switch-memory unit without any additional data.

The controls and displays functional diagram and interface diagram are illustrated in figures A6 and A7, respectively.

TABLE A2. CONTROL AND DISPLAY DESCRIPTION

Device	Slot- channel	Signal type	Bit position	Signal lines	Meaning
HSD	9-1	SED in	0-5	6	0 temperature; 1 = high 1 display status; 1 = fail 2-5 spare
	14-1	SED out	2	1	Power control
VSD	9-1	SED in	6-8	3	6 temperature; 1 = high 7 display status; 1 = fail 8 spare
	14-1	SED out	1	1	Power control
MPD-1	9-2	SED in	0-5	6	0 temperature; 1 = high 1 display status; 1 = fail 2-5 spare
	14-1	SED out	3	1	Power control
MPD-2	9-2	SED in	6-11	6	6 temperature 7 display status 8-11 spares
	14-1	SED out	4	1	Power control
HUD	9-1	SED in	9-11	3	9 display status 10 temperature 11 spare
	14-1	SED out	0	1	Power control
IMFK	9-2	SED in	12-14	3	12 temperature; 1 = high 13 display status; 1 = fail 14 power status; 1 = on
	14-1	SED out	5	1	Power control

TABLE A2. CONTROL AND DISPLAY DESCRIPTION (CONTINUED)

Device	Slot-channel	Signal type	Bit position	Signal lines	Meaning
MFK	9-1	SED in	12	1	12 power status
	14-1	SED out	6	1	6 power control
MMP	9-1	SED in	13	1	13 power status
	14-1	SED out	11	1	11 power control
DEK1	9-1	SED in	15	1	15 power status
	12-2	SED out	9	1	9 activate = 0 Deactivate = 1
	14-1	SED out	13	1	1 power control
DEK2	9-2	SED in	15	1	15 power status
	12-2	SED out	10	1	10 activate = 0 Deactivate = 1
	14-1	SED out	14	1	14 power control
Armament panel	No status bits				
SCU	9-1	SED in	14	1	14 power status
	14-1	SED out	14	1	14 power control

TABLE A3. MULTIPURPOSE DISPLAY GENERATOR DATA

IM slot-channel	Signal type	Word	Bit	Code	Significance
3-1 (MPDG1)	S/D out	1	0-3	1011	Send MPDG/DSMU status
or 4-1 (MPDG2)			4-15		Not used
		2-32	0-15		Not used
		1	0-3	1000	Clear MPDG status Buffer
			4-15		Not used
		2-32	0-15		Not used
		1	0-3	1111	Bootstrap diagnostics
			4-11		Not used
			12-15	0000	MPDG wrap test
				0001	MMU wrap test
		2-32	0-15		Test data to be wrapped
1-4 (MPDG1)	S/D in	6	0		Raster symbol generator failure
2-3 (MPDG2)			1		Stroke symbol generator failure
			13		RT transfer error
			14		RT buffer overload
		17	15		DS/MU command error
		28	10		Loss of interrupt
			13		DMA-MMU channel error
			15		MMU error
		28*	13		RT DMA error
			14		MMU error

*Bootstrap diagnostics only

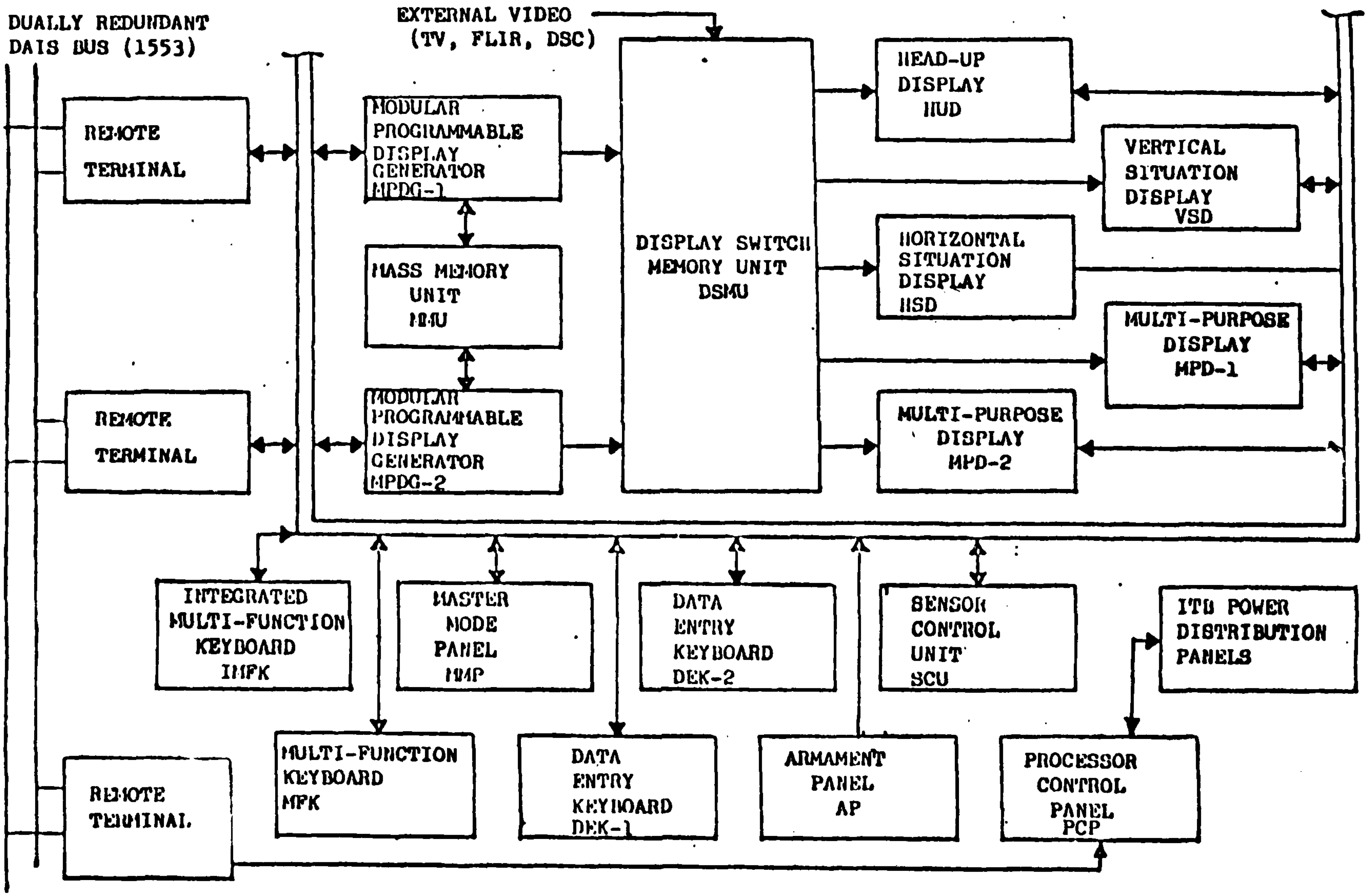

DUALLY REDUNDANT
DAIS BUS (1553)
REMOTE TERMINAL
REMOTE TERMINAL
REMOTE TERMINAL
EXTERNAL VIDEO
(TV, FLIR, DSC)
MODULAR PROGRAMMABLE DISPLAY GENERATOR MPDG-1
MASS MEMORY UNIT MMU
MODULAR PROGRAMMABLE DISPLAY GENERATOR MPDG-2
DISPLAY SWITCH MEMORY UNIT DSMU
HEAD-UP DISPLAY HUD
VERTICAL SITUATION DISPLAY VSD
HORIZONTAL SITUATION DISPLAY HSD
MULTI-PURPOSE DISPLAY MPD-1
MULTI-PURPOSE DISPLAY MPD-2
INTEGRATED MULTI-FUNCTION KEYBOARD IMFK
MASTER NODE PANEL MNP
DATA ENTRY KEYBOARD DEK-2
SENSOR CONTROL UNIT SCU
ITB POWER DISTRIBUTION PANELS
MULTI-FUNCTION KEYBOARD MFK
DATA ENTRY KEYBOARD DEK-1
ARMAMENT PANEL AP
PROCESSOR CONTROL PANEL PCP

Figure A6. DAIS Control and Display Functional Block Diagram

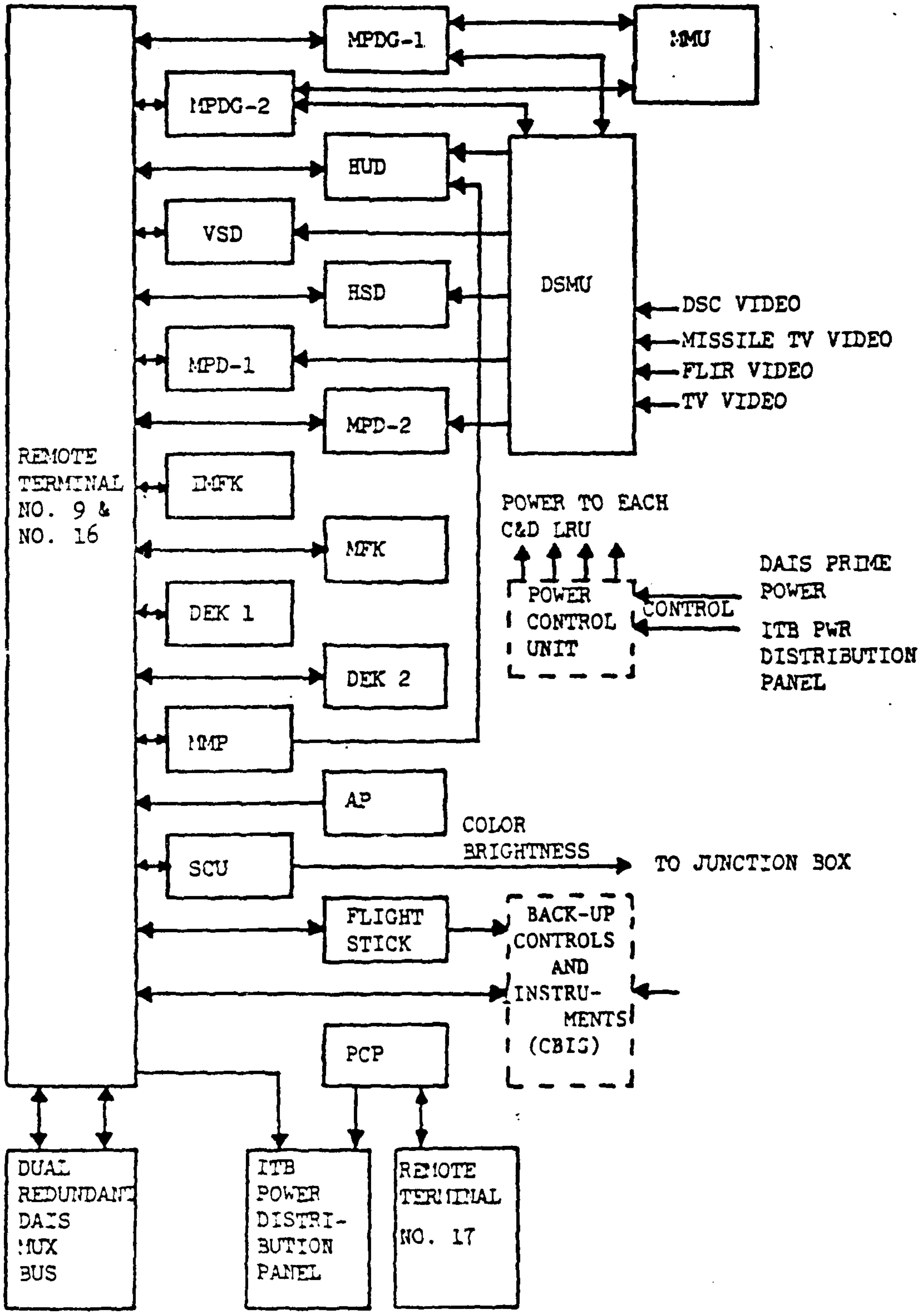


Figure A7. DAIS C&D Interfaces

A.1.4 System Mass Memory

SMM is a support software-implemented core element that contains about 1 million 16-bit words. It contains all data, load modules, and load procedures to load or reload the processors for each system configuration. It can be used to record data for postflight analysis.

The SMM has no defined selftests or BIT word. Their definitions have been left to the developer of the hardware device.

The software implemented version of the SMM does use the status word to detect bus errors and failures in the support equipment. The message error, busy, and terminal failure commands are used.

A.1.5 Processors

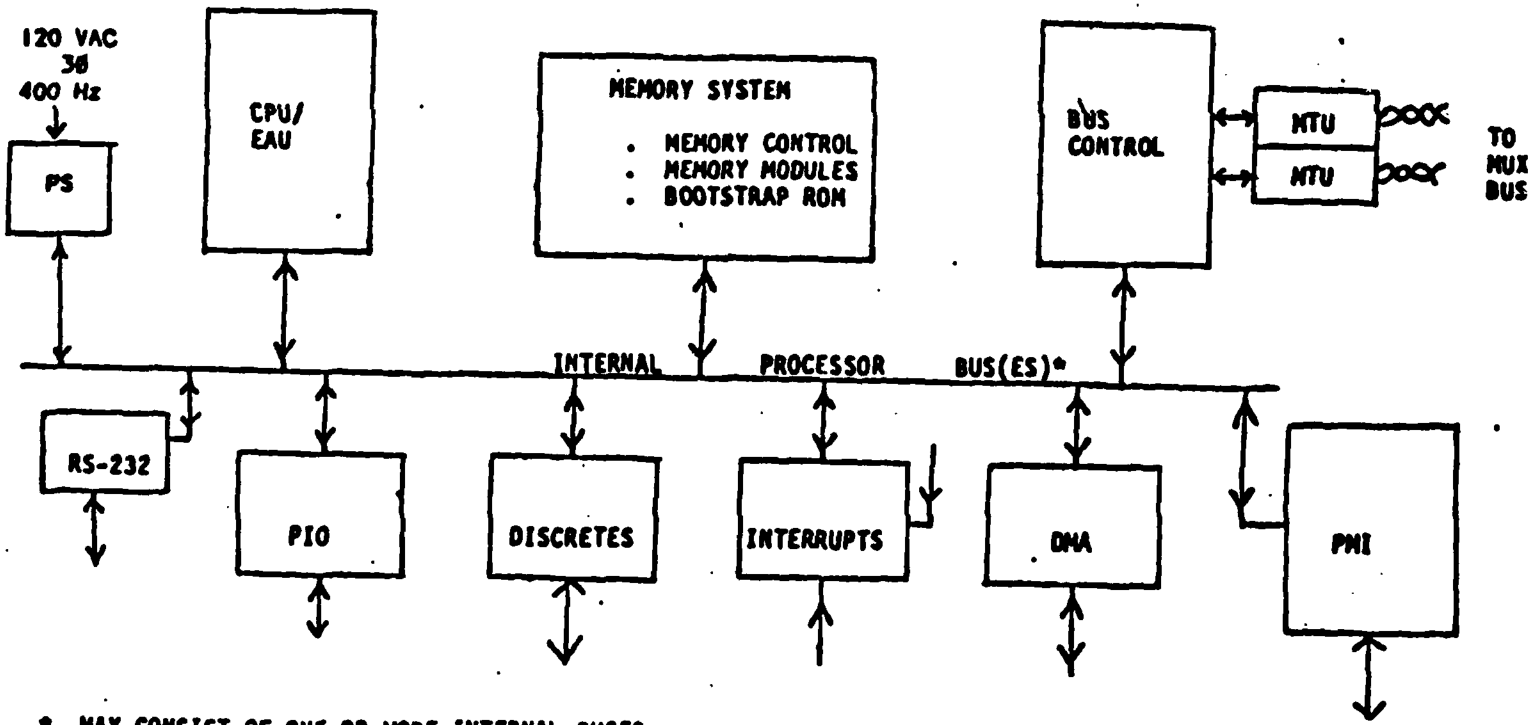
The DAIS processing system consists of four AN/AYK-15A processors. One processor acts as master for the system and controls all bus traffic. The remaining processors operate in remote mode. The integrated bus control function of each processor may act in master or remote mode, depending on software loading.

Functionally, the processor consists of the CPU and memory units with direct memory access to a bus control module (BCM). There are external data interfaces including 16 vectored interrupts, parallel and discrete I/O, and an RS 232C interface. Additionally, a performance monitoring interface (PMI) provides external data and control through processor activity status reporting and a halt line for the processor and its timers. Functional structure is shown in figure A8.

The processor is required to have a minimum of BIT hardware, which translates to a minimum of BIT capability. The maintenance diagnostic calls for a background or pilot initiated self-test routine giving a go/no-go condition with a 90% chance of error detection. Additionally, the processor can detect and report low-voltage and high-temperature conditions. The processor reports self-test failures in the fault register (see fig. A9).

CPU - CENTRAL PROCESSING UNIT
EAU - EXTENDED ARITHMETIC UNIT
ROM - READ ONLY MEMORY
MUX - MULTIPLEX (1553)
RS-232   ANSI RS-232 SERIAL INTERFACE

PIO - PARALLEL INPUT/OUTPUT
DMA - DIRECT MEMORY ACCESS
PMI - PERFORMANCE MONITORING INTERFACE
PS - POWER SUPPLY
MTU - MULTIPLEX TERMINAL UNIT



* MAY CONSIST OF ONE OR MORE INTERNAL BUSES

Figure A8. DAIS Processing System Functional Diagram

0 1	2 3 4	5 6 7	8 9	10 11 12	13 14 15
MEMORY PROTECT	PARITY	I/O	ILLEGAL	SPARE	BITE

The bits will have the following meaning when set to a one (1):

Bit 0 CPU is attempting to write in a protected memory location.

Bit 1 BCI or DMA is attempting to write in a protected memory location.

Bit 2 Memory Parity Error

Bit 3 PIO channel Parity error

Bit 4 DMA channel Parity error

Bit 5 An output command is used with an input OPCODE or an input command is used with an output OPCODE.

Bit 6 PIO transmission error. Other I/O error checking devices, if used, may be OR'd into this bit to indicate an error.

Bit 7 Other I/O errors.

Bit 8 Illegal address; A memory location is addressed which is not present or does not respond

Bit 9 Illegal Instruction; Attempted execution of an instruction whose first 16 bits are not defined by this document shall cause this bit to be set. Undefined bit fields in the first 16 bits of an instruction are reserved.

Bit 10-12 Spare for future use, presently undefined.

Bit 13 Hardware built in test equipment (BITE) error is detected.

Bit 14-15 These bits are for use by the designer for further defining (coding, etc.) the BITE error which is detected. This can be used with Bit 13 to give a more complete error description. If minimal or no BITE is designed in, these bits will be set to zero.

Figure A9. DAIS Processor Fault Register Format

The BCM monitors bus I/O for errors. It actively maintains message error data in the same manner as a remote terminal. Additionally, in quiescent mode, the master controller can self-test by sending a predefined message to its own address, monitoring and performing validity checks on the incoming data. The remote processor can be commanded by mode command 3 to perform self-tests. The BCM BIT word is shown in figure A10.

A.1.6 Stores Management

The stores management function is simulated through DEC-10 software. There is no BIT function, but system status can be monitored through the serial digital input stream. This stream contains BITs for status of relays, weapons presence, firing circuit unlocked, power on, and power level ok.

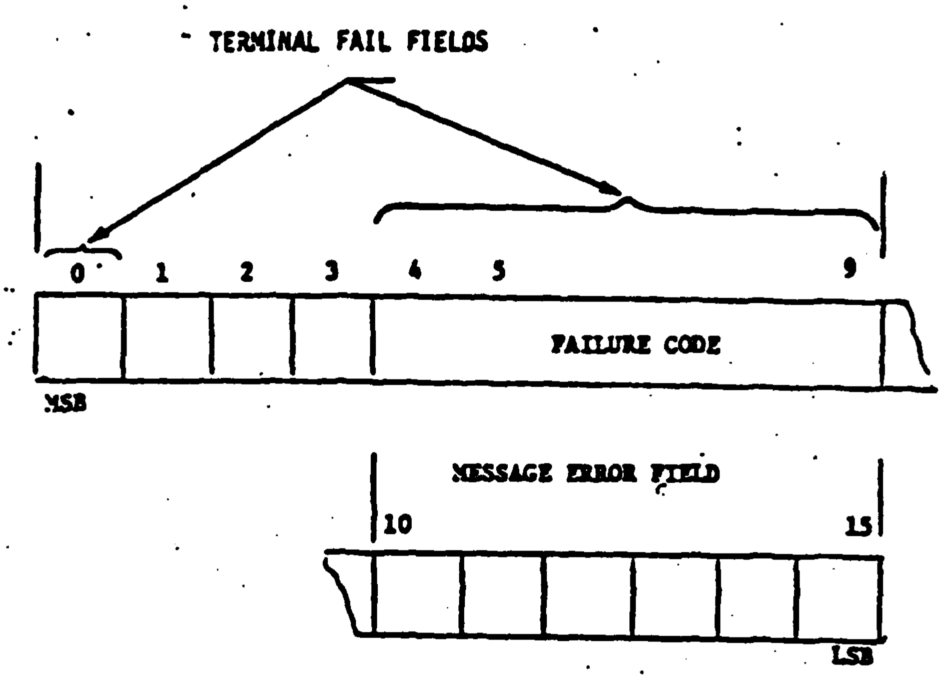
A.1.7 Pilot and Crew

The current pilot and crew interaction with testing and maintenance functions is through the processor control panel (PCP) which can be used to start, restart, and load to system hardware and software. He can also initiate a processor self-test routine per part 1, mode 1 of the processor maintenance diagnostic manual.(19)

A.2 Status of Testing and Maintenance Subsystems

Testing and maintenance concepts are for the most part unimplemented in the DAIS system. However, system structure allows implementation with minor modifications in RT firmware. Operational software is developed for the mission processor, and the DEC-10 sensor simulation software is modified so that faults could be simulated in the avionics.

The current test system is implemented in the master processor executive. It consists of a tallying of message error and terminal fail flags received in status words over the MIL-STD-1553B bus. No response and other anomalous conditions are handled via the system control procedures defined in the System Control Procedures(20) specification.



BIT	MEANING
0	POWER ON RESET
	SPARE
	MTU #1 OUT
	MTU #2 OUT
	SPARE
	BCM/MTU 1 SELF TEST
	BCM/MTU 2 SELF TEST
	SPARE
	SPARE
	SPARE
	NO DATA RECEIVED
	WORD COUNT HIGH
	WORD COUNT LOW
	DATA PARITY ERROR
	INVALID DATA
15	INVALID COMMAND

Figure A10. Bus Control Module BIT Word Format

A.3 Recommendations for DAIS Implementation of ITM

For the baseline system, ITM functions should be implemented in the master processor applications software. Using existing capabilities of LRU's in the DAIS, a sufficient baseline demonstration could be performed. No hardware will be modified. However, as described later, firmware in the remote terminal will be modified to allow acquisition of BIT data from multiple remote sensing units through one acquisition command. A summary of recommendations for the baseline system and rationale for each follows.

a. Reprogram remote terminal EROM'S—the benefit of this action is to gain single-command access to all BIT data for devices with interfaces to nine remote terminals. The EROM'S are firmware that is used to implement a table of sequential RT channel accesses. Any sequential set can be formed into a single MIL-STD-1553B message (up to 32 words) thereby reducing bus overhead. Presently, BIT data available at the remote terminal are not organized into a single access group.

   An alternative considered was the implementation of the subsystem status BIT in the MIL-STD-1553B status word. The remote terminal received status BITs from the sensors and logically OR them together to form a single BIT in the status word. The BIT is one if any device reported a fault, and ITM would perform diagnostics oy asynchronous data acquisitions from the remote terminal. This alternative was rejected because (1) the expense of hardware modifications in the remote terminal does not afford a significantly faster data acquisition rate, and (2) the system loading when a fault occurs is greater for this scheme than for the one selected.

b. Upgrade simulated BIT capability for simulated sensors—the DAIS laboratory implementations of several sensors contain no BIT data. These simulations (ILS, radar altimeter, UHF, VHF, PAVE TACK) should be upgraded so they can simulate common failures. This allows better exercising of ITM in the testbed and aids in determination of the adequacy of BIT schemes for these devices.

c. Device software tests to supplement BIT in core elements—processor, controls, and displays specifications require only a 90% level of fault detection. They are hardware elements and therefore are not easily modified. Software testing through interleaved BIT could boost fault detection to a reasonable level, perhaps 95%. This allows simulation of BIT performance expected on next-generation equipment.

d. Master processor executive should be modified to support ITM operations—the current DAIS executive performs most of the logging and diagnostic functions in the current system. Because of its size, ITM should be implemented as an applications task. This will require that some modifications to the executive and to the executive-applications task interface.

The executive must have a mass memory logging function that will support applications tasks and postflight processing. New service requests must be implemented in the executive-to-applications interface for reading and writing the mass memory devices.

The executive-to-applications interface must be modified to allow communication of status information. The applications tasks must have access to all BIT data from the buses and LRU's in the system. T.e applications task must be able to flag a device as failed and discontinue executive communications with the failed device. The system status must be available from ITM applications tasks at the pilot's request.

e. Applications task distribution—applications tasks should be distributed between mission processors to reduce loading on the master processor. Only a small core of functions should be in the master processor—those that require access to data transmitted in the MIL-STD-1553B status word and mask command responses. Special test technique models should be distributed to other mission processors and should report to the ITM core only in the event of a test failure.

APPENDIX B

HARDWARE CONFIGURATION MATRIX AND FUNCTIONAL FLOW DIAGRAMS FOR THE ADVANCED SYSTEM FUNCTIONAL SUBSYSTEMS

Table B1. Navigation Hardware Configuration Matrix

Hardware function →	ICNIA - NAV SECT.												
Operating Modes ↓	ILS/VOR	TACAN	GPS	JTIDS	INS	ADAM	HARS	TERCOM	AIR DATA	TF/TA	RADAR ALT	NAV FLIR	SAR
Integrated Navigation	glide slope	r b	p	TOA	p,V at ac	p		p	s	alt	alt		
GPS			p v		at		at						
Inertial Navigation					p,v at ac								
Area Navigation		r b					at						
Dead Reckoning							at		s				
JTIDS Relative Navigation				p,v at									
Terrain Complete						p	at			alt	alt		
Position Update								p			alt	td	td

LEGEND

ac	acceleration	b	bearing	s	air speed	v	velocity
alt	altitude	p	position	td	terrain data		
at	attitude	r	range	TOA	time of arrival		

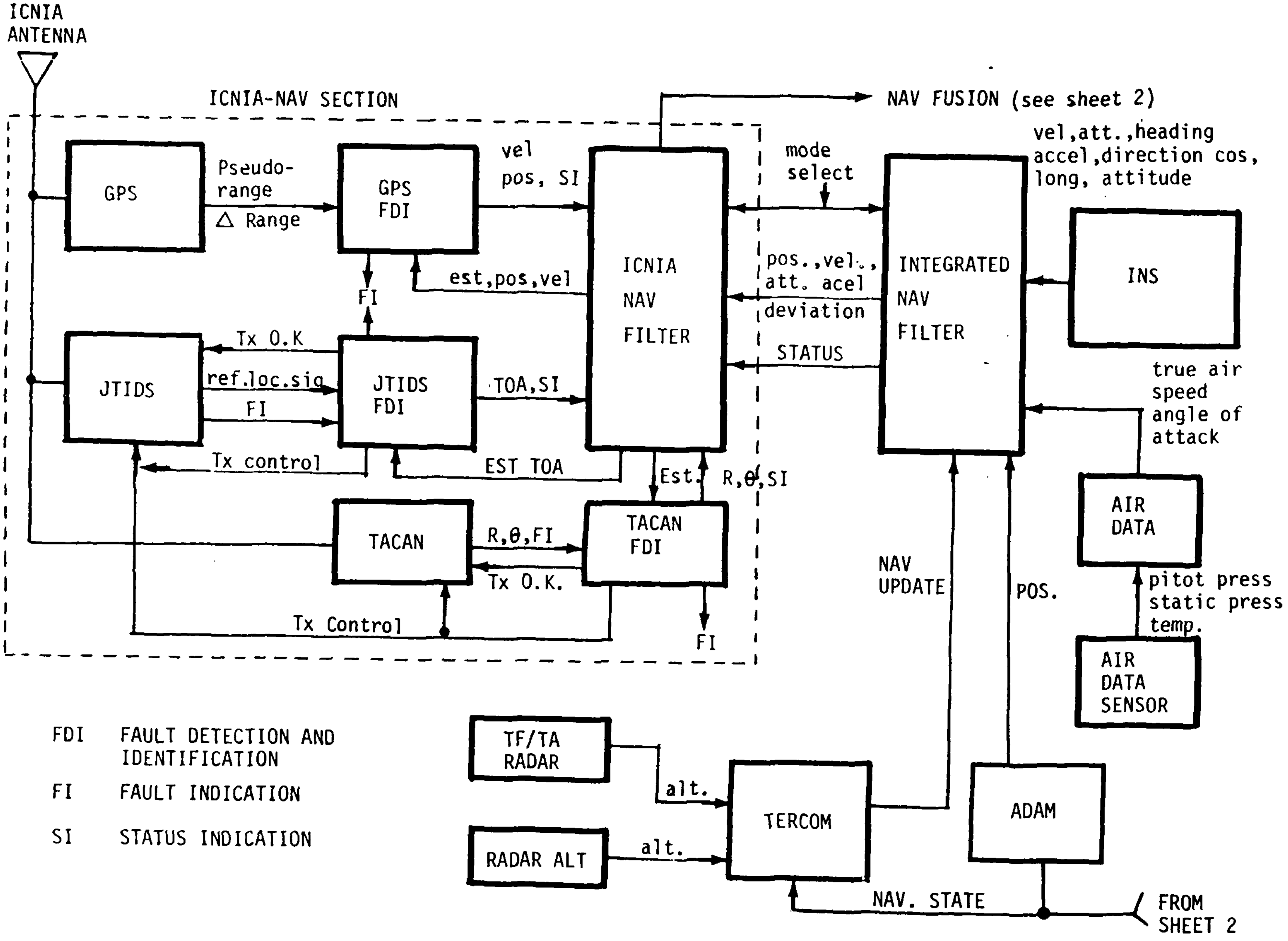


Figure B1. NAV System Functional Flow - Integrated NAV Mode

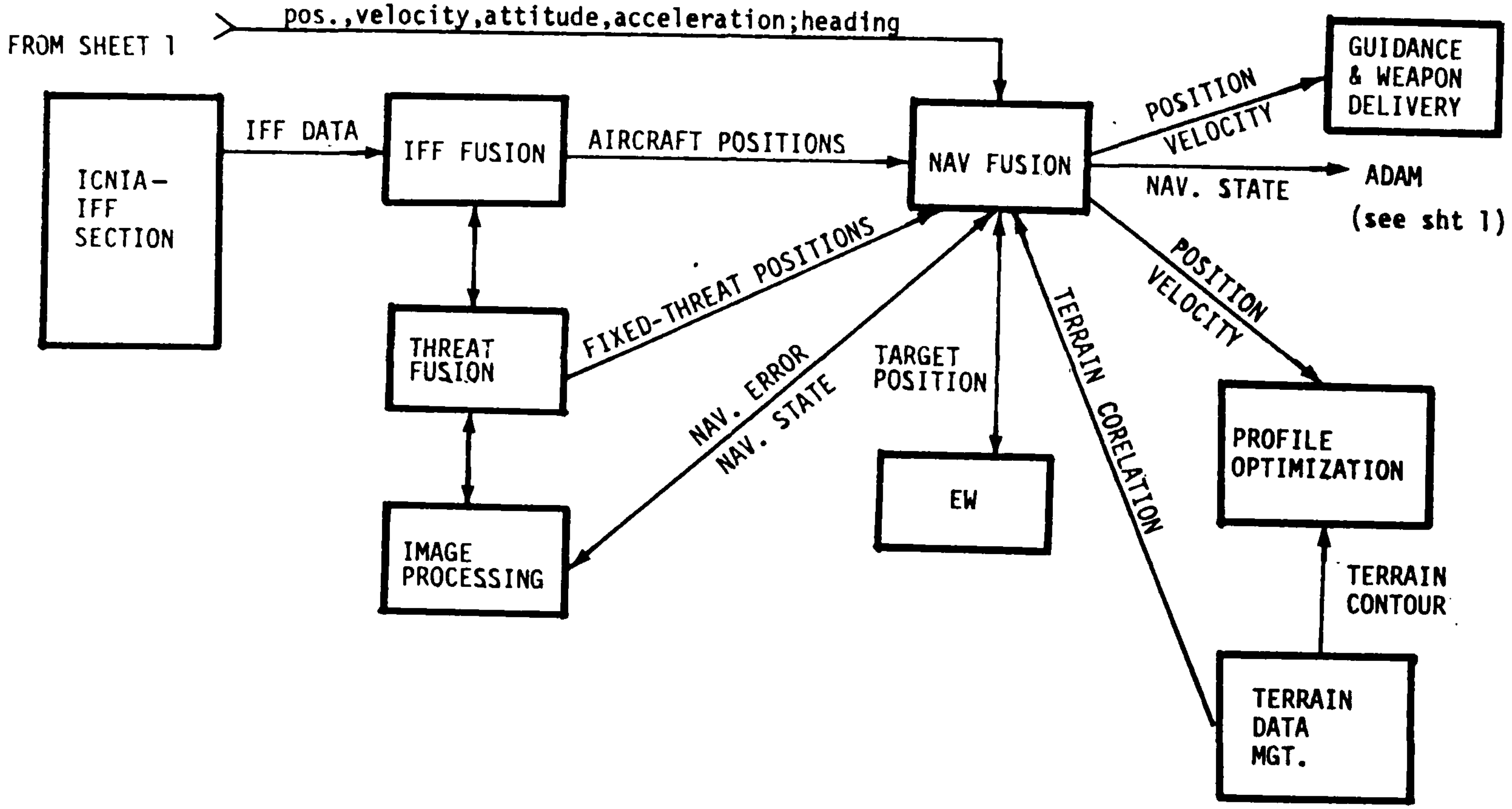


Figure B1. Nav System Functional Flow - Integrated Nav Mode (Continued)

TABLE B2. GUIDANCE/FC HARDWARE CONFIGURATION MATRIX

HARDWARE / FUNCTION / MODES	CONTROLS & DISPLAYS	ICNIA				TERCOM	ADAM	TF/TA	APMS
		IFF	GPS	TACAN	ILS				
INTEGRATED GUIDANCE		threats				terrain contour	route	alt, terrain hazzard	threat list
TF/TA						way-point alt	way-point coord.	alt	
COMMAND NAV							way-point coord.		
4-D NAV/ AUTOTHROTTLE							way-point coord.		
COMMAND HEADING	heading input								
COMMAND TRACK							route		
AUTO POP-UP		threats	a/c pos.				target coord.		threat coord.
TACAN				route					
ILS STEERING					glide slope				
TERRAIN MAP						terrain contour	route	terrain hazzard	
ATTACK GUIDANCE							route		threat coord.

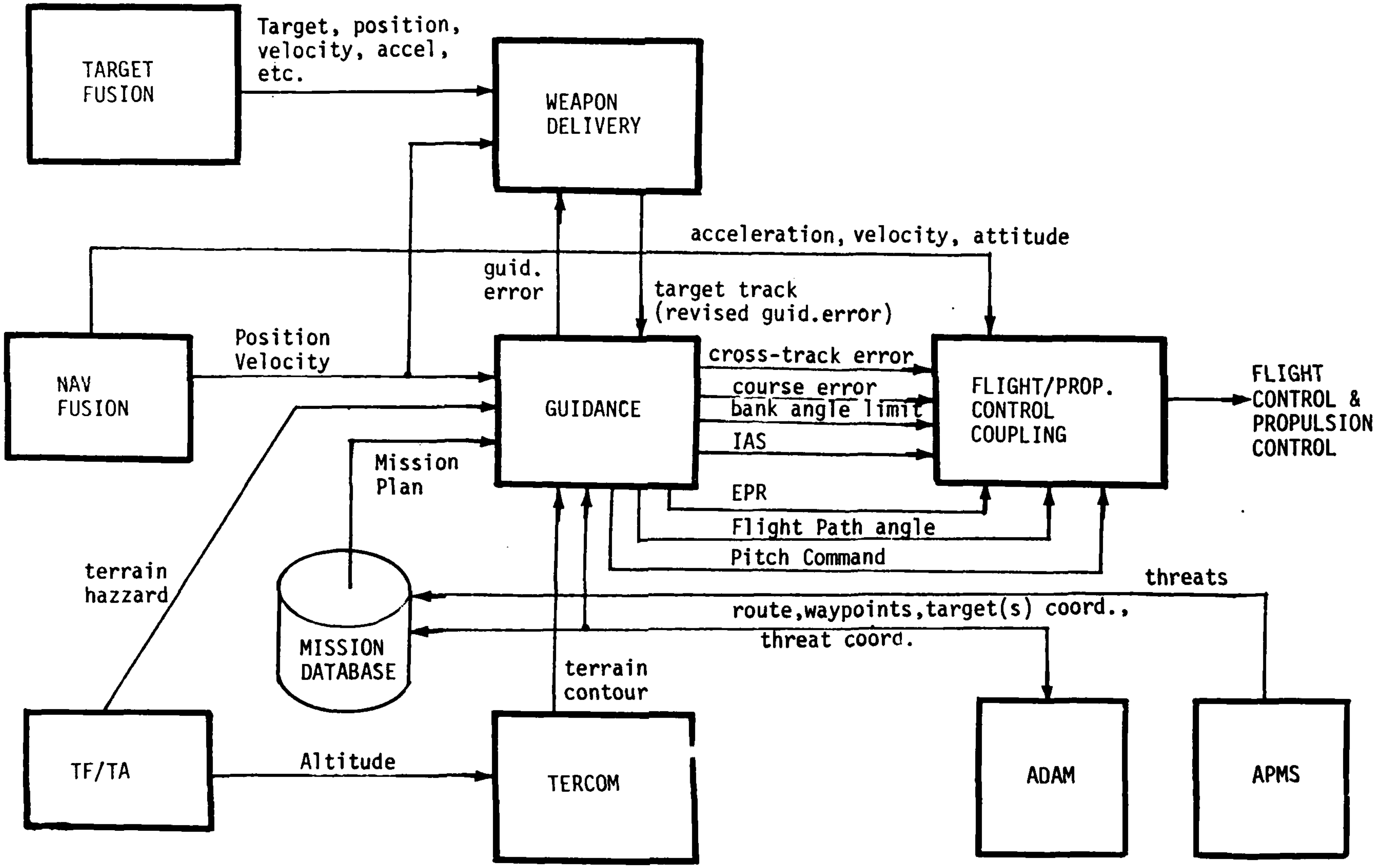


Figure B2. Guidance/Flight Control Functional Flow

TABLE B3. WEAPON DELIVERY HARDWARE CONFIGURATION

OPERATING MODE / HARDWARE FUNCTION	IRR MAVERICK	MMW	LASER DESIGNATOR	LASER ILLUM/RANGER	LASER DET/TRACK	NAV BOMB	HUD/FLIR	SAR	ADAM	TARGET FUSION	JTIDS	NAV FUSION	IMAGE FUSION	THREAT FUSION	GUIDANCE	PROFILE OPTIMIZATION	STORES MANAGEMENT	FLIGHT CONTROLS	MISSION DATABASE	C/D	HARS
IFFC AUTO		OPTIONAL RANGE & BEARING	ILLUMINATION	RANGE	DETECTION TRACKING			RANGE & BEARING	TERRAIN PROFILE	TARGET DATA	TARGET POSITION	VELOCITY ACCEL ALT.	TARGET RECOG. LOC.	THREAT LOCATION	BANK IAS,EPR ANGLE PITCH	MISSION PROFILE	STATUS ARM FUSE	STEERING DATA	MISSION		
IFFC MANUAL		OPTIONAL RANGE & BEARING	ILLUMINATION	RANGE	DETECTION TRACKING			RANGE & BEARING	TERRAIN PROFILE	TARGET DATA		VELOCITY ACCEL ALT.	TARGET RECOG. LOC.	THREAT LOCATION	BANK IAS,EPR ANGLE PITCH	MISSION PROFILE	STATUS ARM FUSE		MISSION		YAW PITCH ROLL
CCIP AUTO		OPTIONAL RANGE & BEARING	ILLUMINATION	RANGE	DETECTION TRACKING			RANGE & BEARING	TERRAIN PROFILE	TARGET DATA		VELOCITY ACCEL ALT.	TARGET RECOG. LOC.	THREAT LOCATION	BANK IAS,EPR ANGLE PITCH	MISSION PROFILE	STATUS ARM FUSE			OFFSET AIM POINT	YAW PITCH ROLL
CCIP MANUAL		OPTIONAL RANGE & BEARING	ILLUMINATION	RANGE	DETECTION TRACKING		AIMING RETICLE	RANGE & BEARING				VELOCITY ACCEL ALT.	TARGET RECOG. LOCA		BANK IAS,EPR ANGLE PITCH		STATUS ARM FUSE				YAW PITCH ROLL
AGM AUTO	VIDEO IMAGE/ TRACKING							RANGE & BEARING		TARGET DATA		VELOCITY ACCEL ALT.	TARGET RECOG. LOC.		BANK IAS,EPR ANGLE PITCH		STATUS ARM FUSF			OFFSET AIM POINT	
AGM MANUAL	VIDEO IMAGE/ TRACKING											VELOCITY ACCEL ALT.			BANK IAS,EPR ANGLE PITCH		STATUS ARM FUSE			OFFSET AIM POINT	
NAV BOMB						TARGET COORD. NAV. DATA						VELOCITY ACCEL ALT.					STATUS ARM FUSE				YAW PITCH ROLL
WASP (TBD)																					
HYPERVELOCITY (TBD)																					

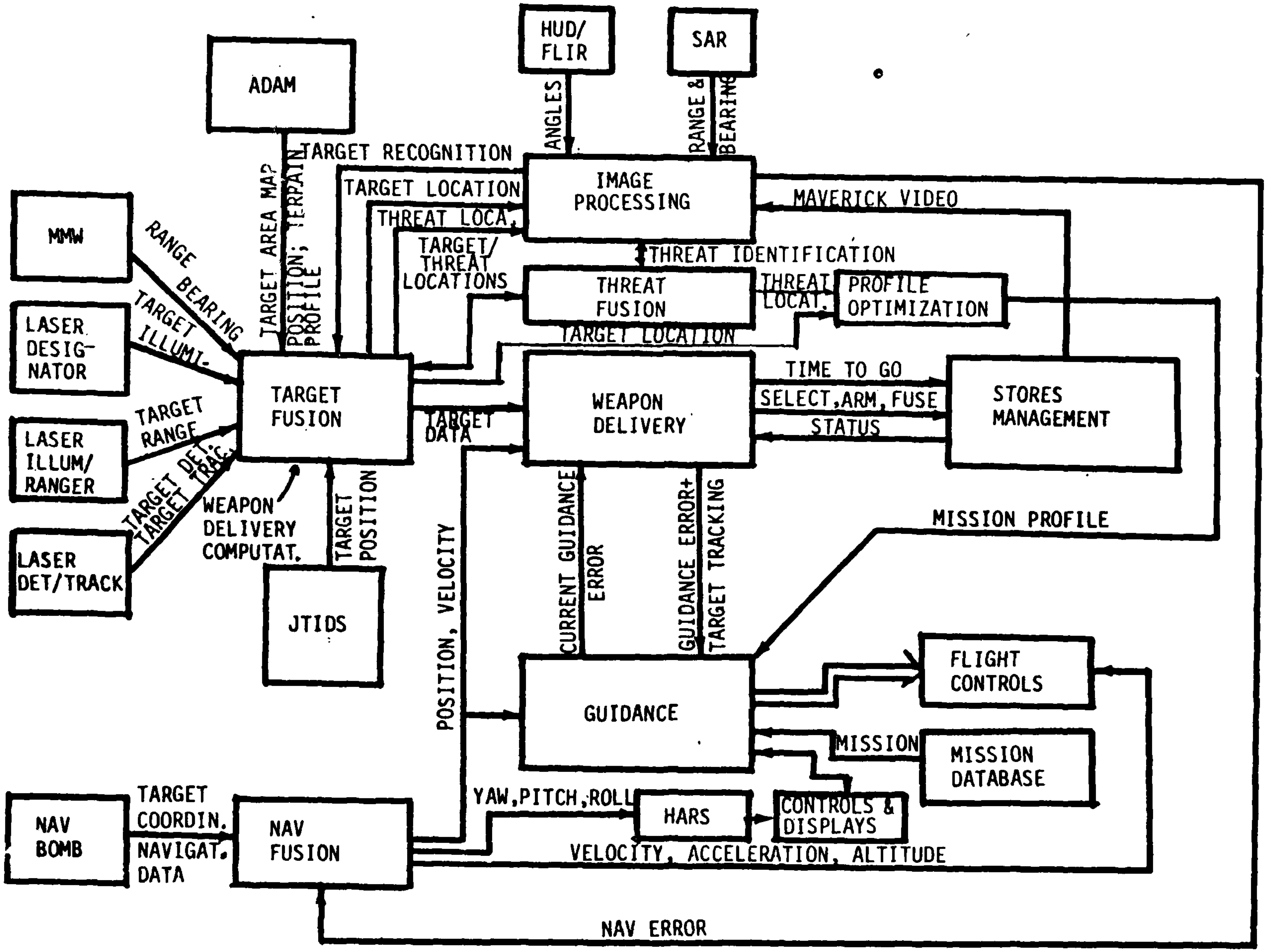


Figure B3. Weapon Delivery Functional Flow

TABLE B4. TARGET ACQUISITION HARDWARE CONFIGURATION MATRIX

	MULTISENSOR SYSTEM															
	LASER DESIGNATOR	NAV FLIR	LASER DET/TRACK	LASER ILLUM/RNG	ATTACK FLIR	TARGET SCREENER	IIR MAVERICK	SAR	MMW	JTIDS	TARGET FUSION	IMAGE PROCESSOR	PROFILE OPTIM.	THREAT FUSION	ADAM	TERCOM
AUTOMATIC	TARGET ILLUMINATION	WIDE FOV IMAGE	AUTOMATIC TARGET TRACK	TARGET RANGE	NARROW FOV IMAGE	TARGET CLASS; TRACKING SIGNAL	VIDEO IMAGE/ TRACKING	HIGH RESOL. RADAR IMAGE	ADVERSE WEATHER IMAGE/ TRACKING	TARGET POSITION	TARGET DATA	TARGET RECOGNITION	MISSION PROFILE	THREAT LOCATION	TARGET AREA MAP	TERRAIN CONTOUR
MANUAL		WIDE FOV IMAGE			NARROW FOV IMAGE											

FOV = FIELD OF VIEW

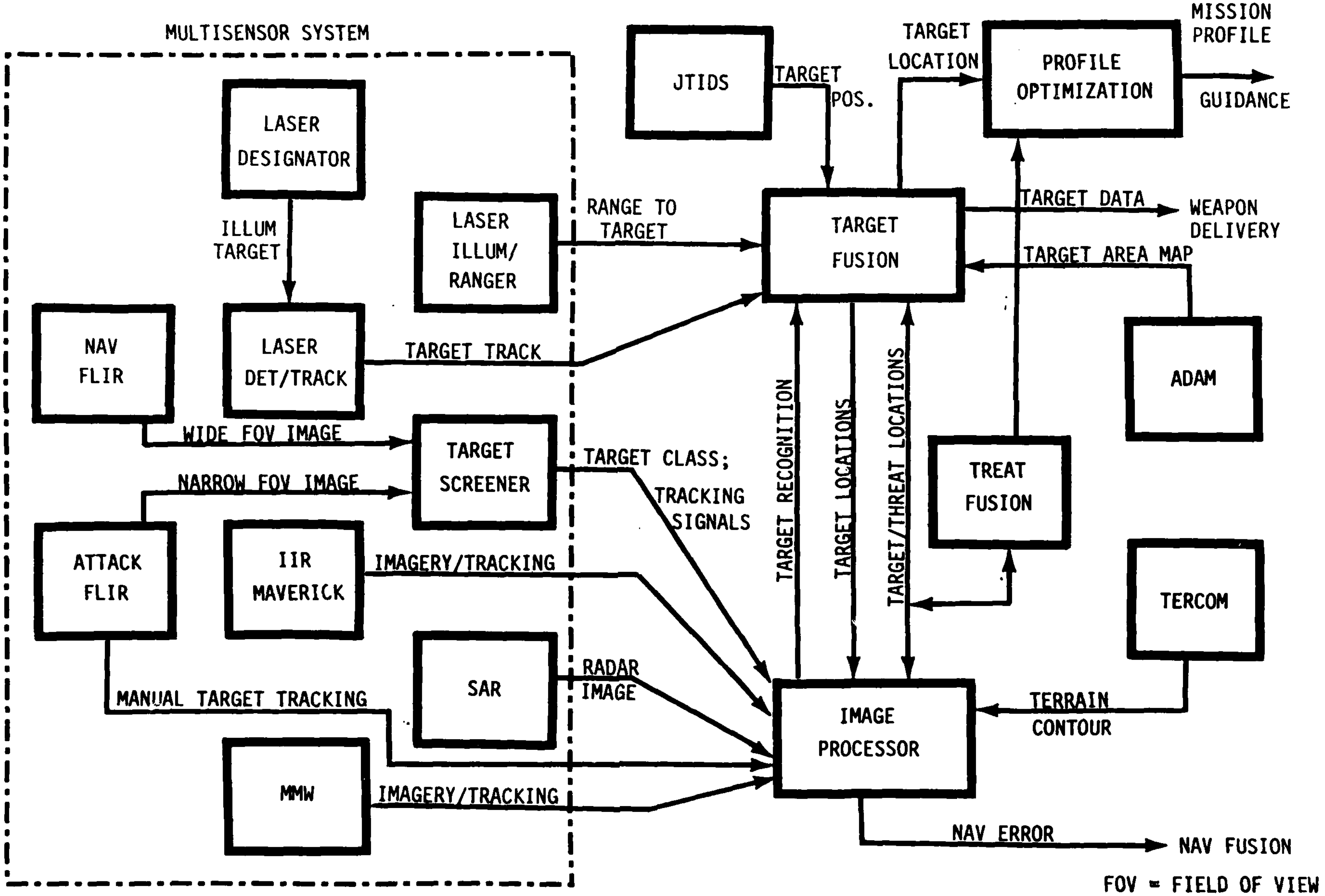


Figure B4. Target Acquisition Functional Flow

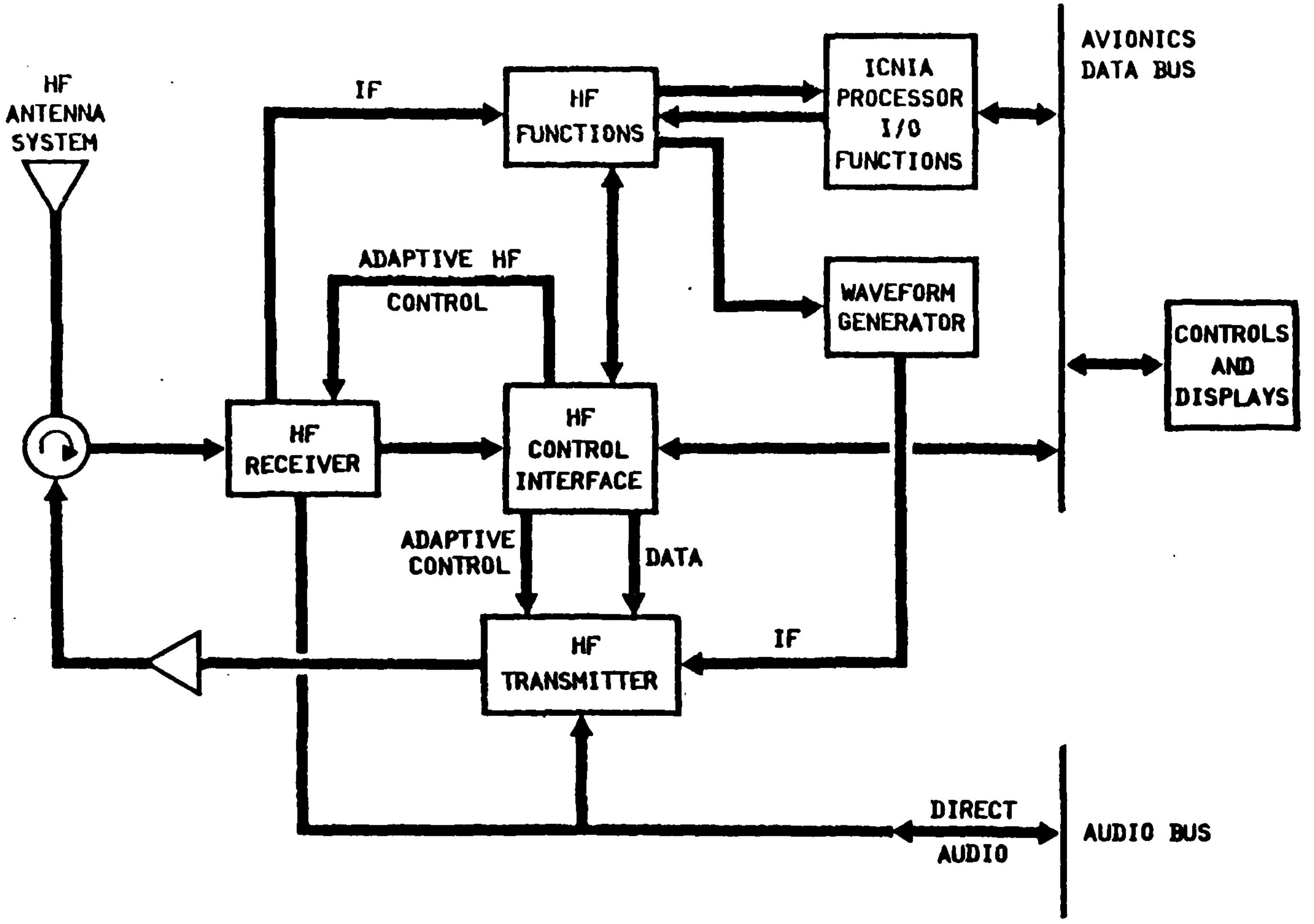


Figure B5. Communications Functional Flow (ICNIA HF)

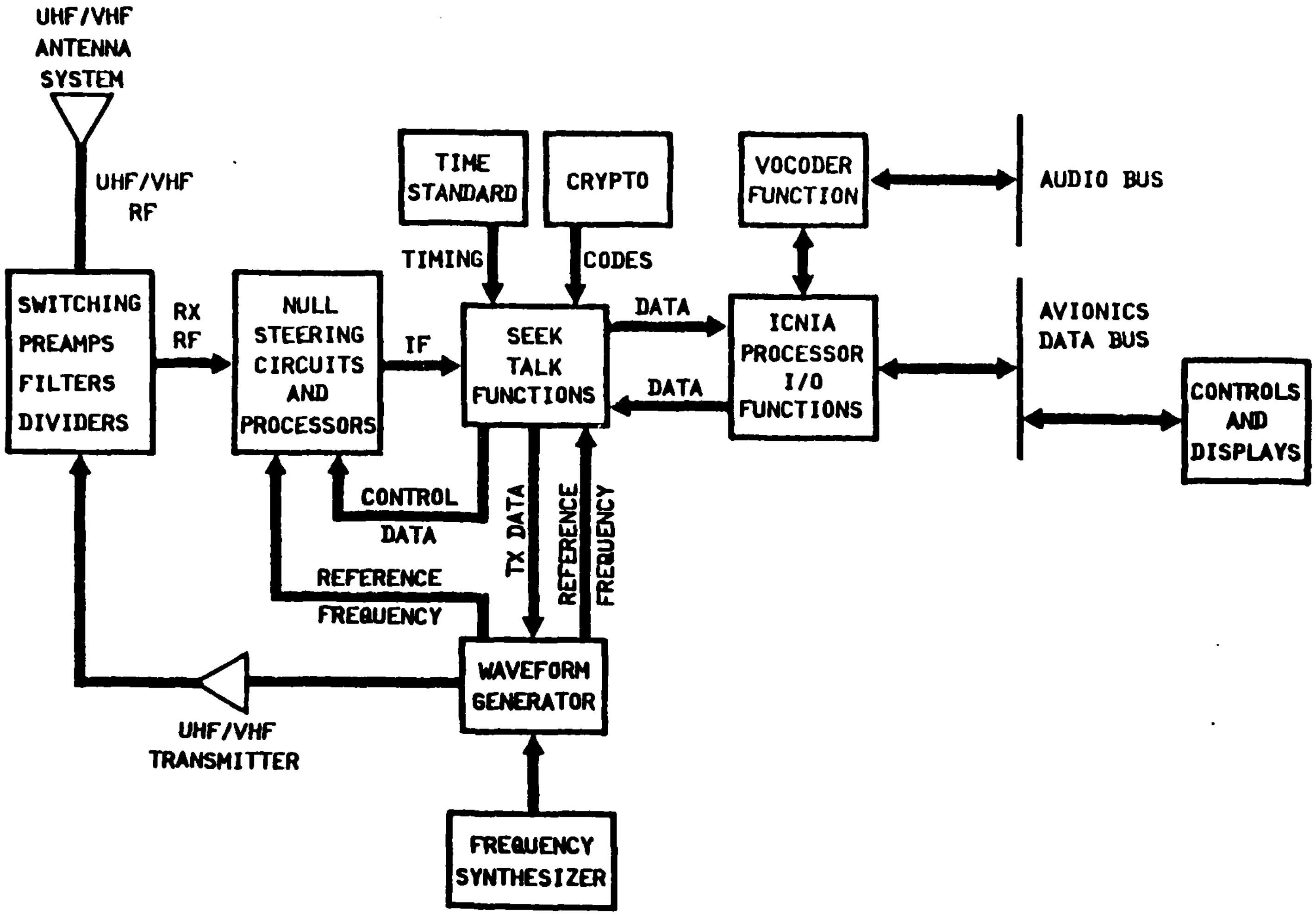


Figure B6. Communications Functional Flow (ICNIA UHF/VHF)

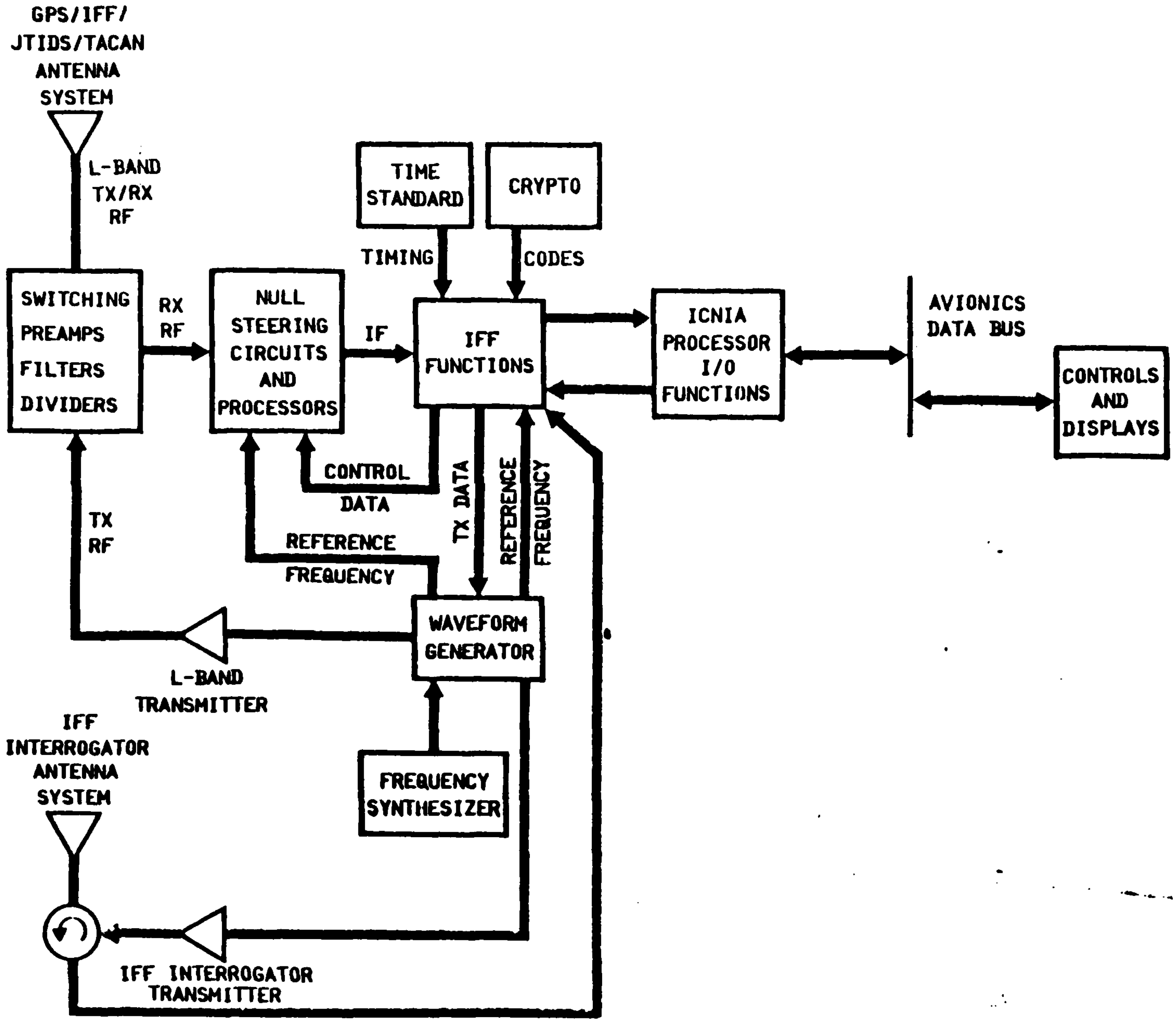


Figure B7. Communications Functional Flow (ICNIA IFF)

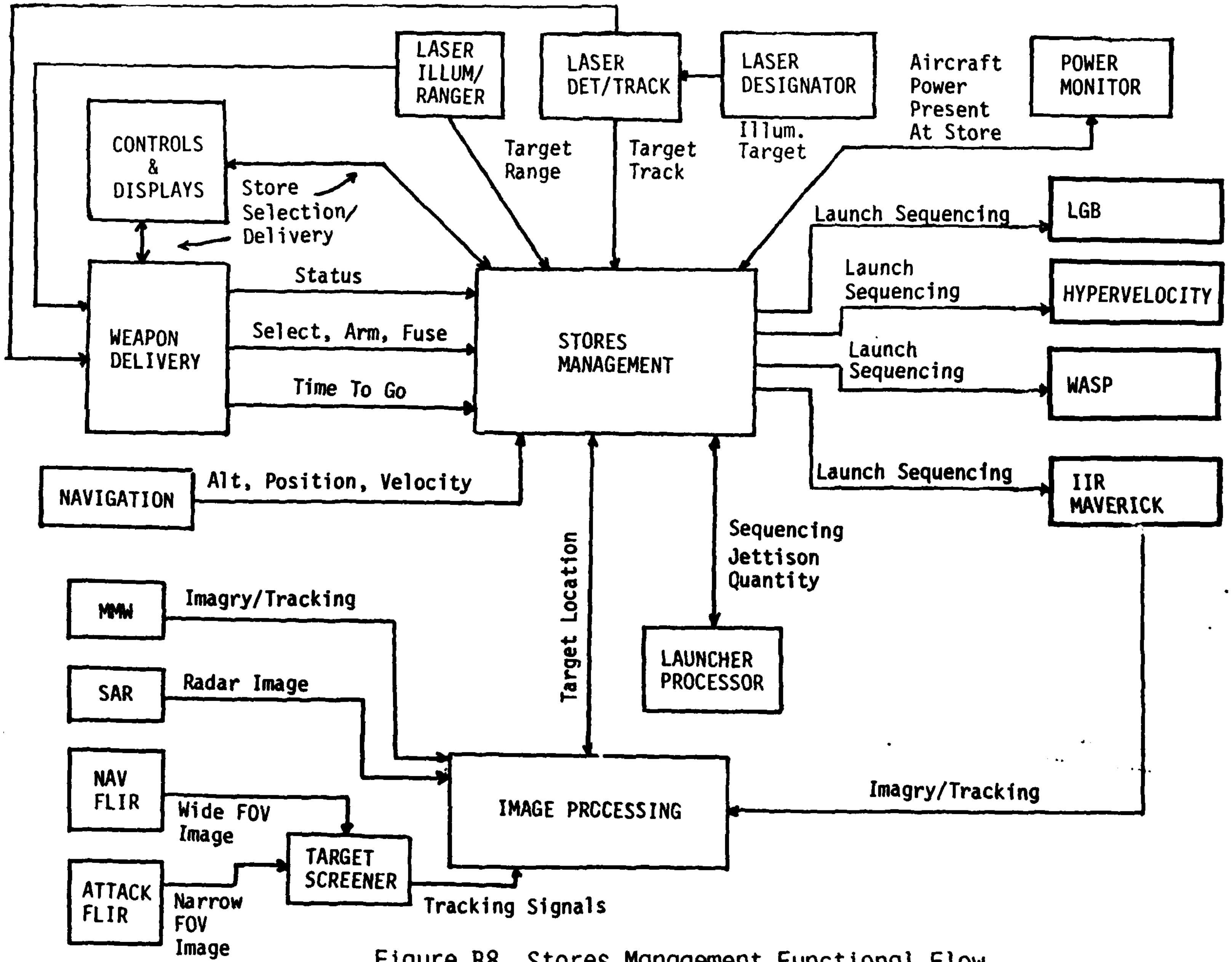


Figure B8. Stores Management Functional Flow

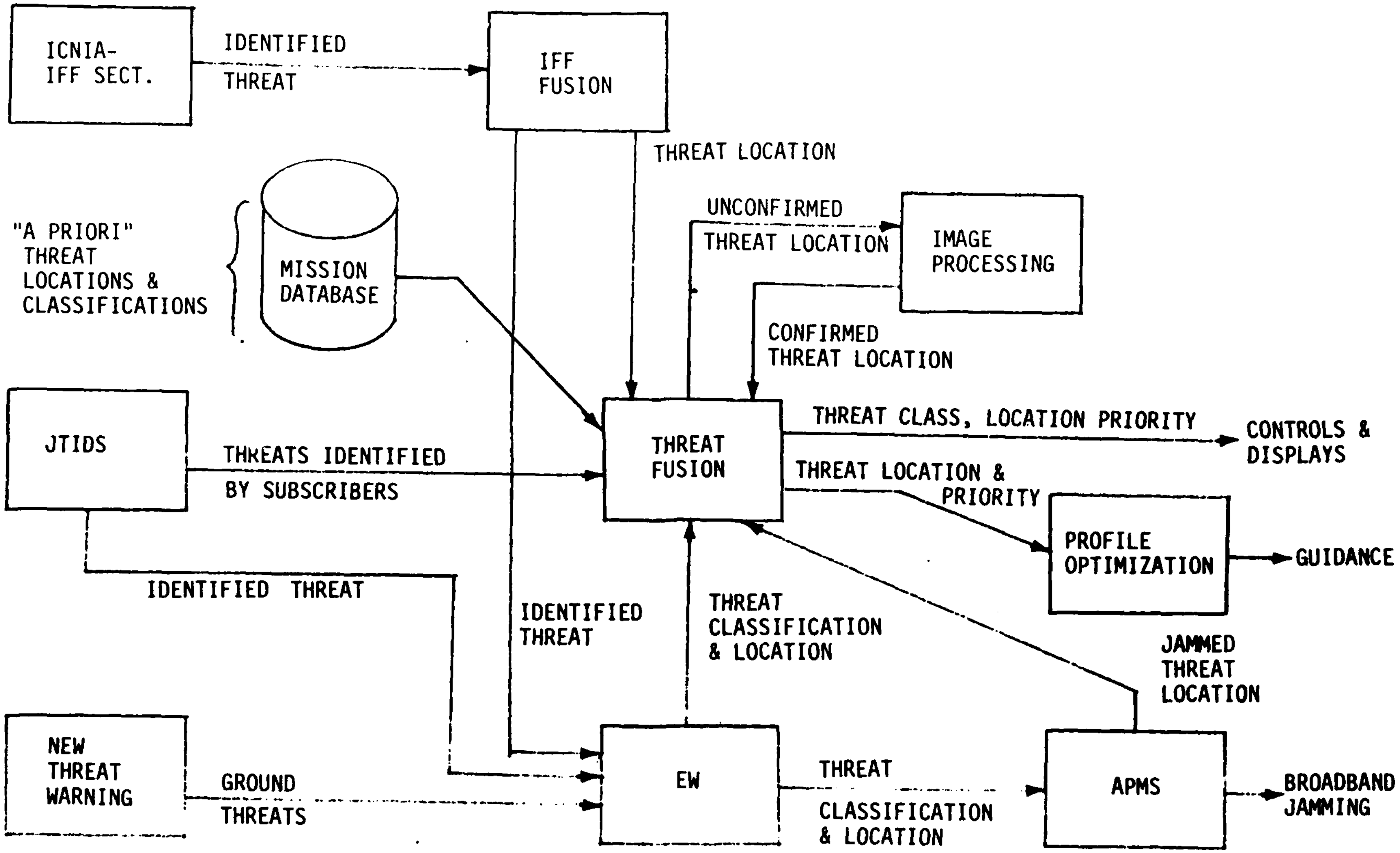


Figure B9. Threat Management Functional Flow

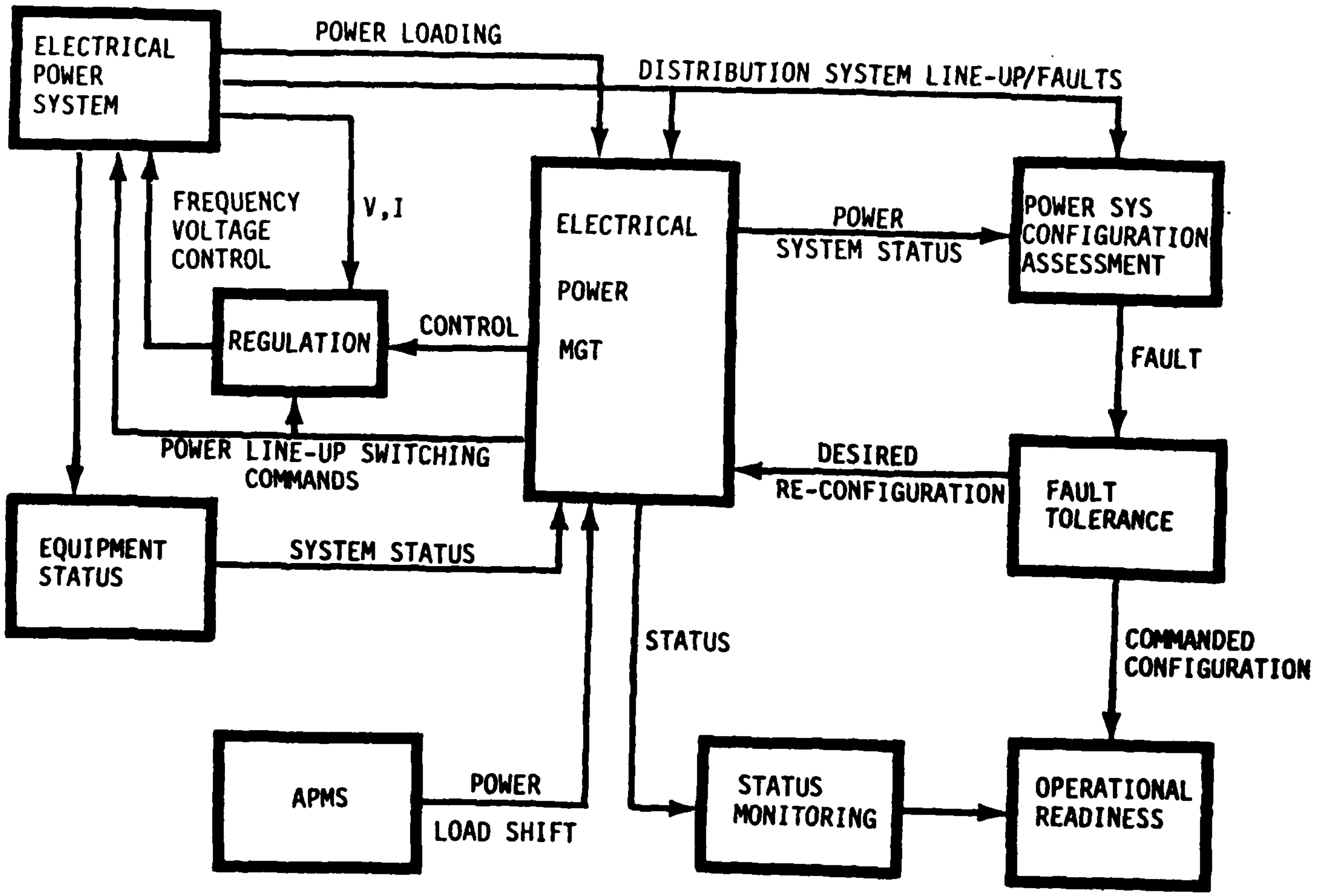


Figure B10. Electrical Power Management Functional Flow

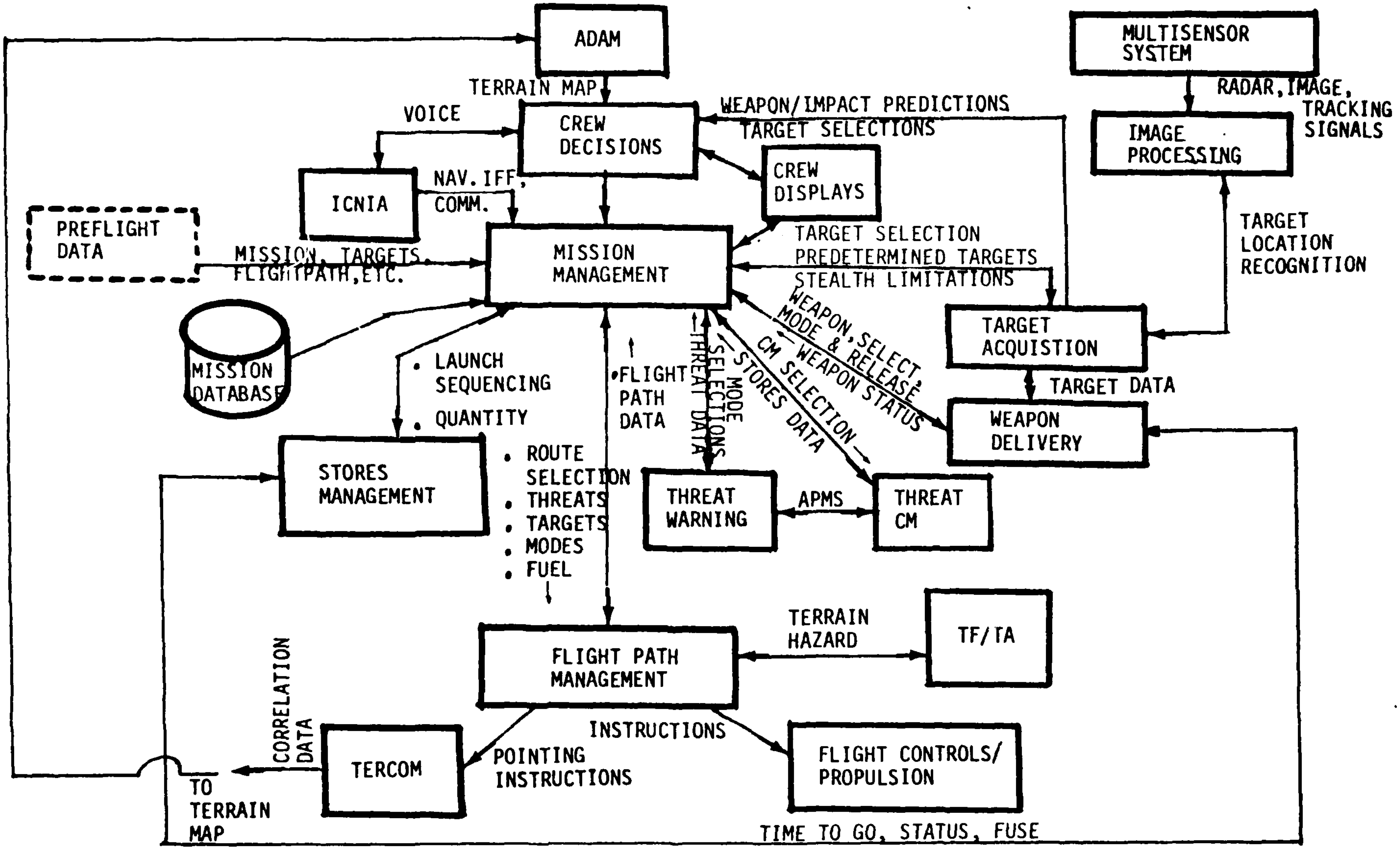


Figure B11. Mission Management Functional Flow

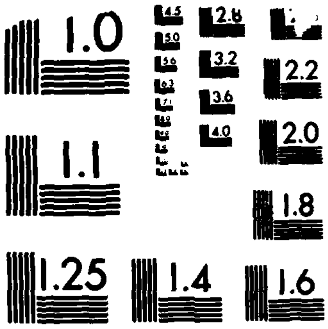
APPENDIX C

SIZING AND TIMING

AD-A138 587 INTEGRATED TESTING AND MAINTENANCE TECHNOLOGIES(U) 4/4
BOEING AEROSPACE CO SEATTLE WA R O DENNEY ET AL.
DEC 83 AFWAL-TR-83-1183 F33615-81-C-1517

UNCLASSIFIED F/G 5/1 NL

END
FILMED
4-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TABLE C1. SIZING AND TIMING ESTIMATES

Program storage

Function	Description	Size (word = 16 bits)
Bit data collection data		
o Limit data	o Upper and lower limits for three tests and 70 LRU's	420
o Status words	o Incoming values from 70 LRU's	210
o Entries in synchronous I/O tables	o One status word from 20 smart addresses: Six status words from one RT	26
	o One entry per LRU per limit + one entry per status request	91
o Overhead		150
Bit data collection-processing		
o Status checking	o Compare 26 status words to nominal values	70
o Limit checks	o 70 LRU's upper and lower limits	30
o Overhead		150
Reasonableness testing	o 14 reasonableness tests in mission processor	150
	o Compute average	
	o Test to limits	
	o Voting	
	o Loop control	
	o Compute rate of change	
	o Collect reasonableness test results from fusion processors	
	o Tables	
	o Average	3,000
	o Limits	200
	o Rate of change	500

TABLE C1. SIZING AND TIMING ESTIMATES (CONTINUED)

Program Storage

Function	Description	Size
Failure occurrence threshold testing	Test threshold for occurrence of test or box failure	50
Emergency (time critical) handling procedures	Check emergency handling table to determine if critical situation*	10
	512 tests (32 words) by 10 flight phases	320
	Defer continuation of check until critical time over	20
	Jump to start of emergency procedures	30
Fault status	Send failed/degraded device or subsystem code to fault tolerance	200
Reconfiguration verification	Receive results of fault tolerance and update backup table	80
Component control	Direct ITM actions at system level	600

*Emergency procedures for each flight phase are part of non-ITM OFP

TABLE C1. SIZING AND TIMING ESTIMATES (CONTINUED)

Program Storage

Function	Description	Size
Statistical tests	Iterative	
	Whiteness test (algorithm and data storage)	300
	Full whiteness test (algorithm only)	800
	Covariance signatures (algorithm)	700
	Sample mean	480
	Pattern recognition	700
Self-test	Decode command	250
Initialization	Determine if interruptive S/T possible	250
	Initiate S/T	250
	Receive results	250
	Encode and send to display	250
Status request	Decode command	200
	Format status data	200
Suspicion	Decode command	290
Identification	Determine test data requested	290
	Format data for C&D	290
	Send message	

TABLE C1. SIZING AND TIMING ESTIMATES (CONTINUED)

Program storage

Function	Description	Size
System status maintenance	For 70 LRU's and 11 subsystems, data available will be:	
	bits 00 go/on-line	350
	01 go/off-line	
	10 no-go/off-line	
	11 performing initiated test	
	Mode of operation	
	Degradation descriptor	
	Computation of status algorithms	
	Failure status for system (MESL)	200
	Failure status for subsystem	150
	Failure status for LRU	100
	Test correlation and LRU correlation update	200
	Mass memory address compute	150
Data recording	List of data per subsystem fault or more change	600
	Acquire data	300
	Send to write only mass memory	110
Fault verification	Initiate failed test	30
	Check results for repeat of error	20
Fault isolation (active diagnostics)	Fault free operation algorithm	450
Fault isolation (correlation of fault records)	Algorithm for performing correlation tests with pertinent operational modes, environmental data, etc.	550

TABLE C2. THROUGHPUT (FAULT-FREE ENVIRONMENT, NO PILOT INTERACTION)

Function	Description	OPS
Collect bit data	o 3 memory fetches and 2 compares for 70 LRU's—limit tests	420
	o 2 memory fetches and 1 compare for 70 LRU's—status check	140
	o Loop control	20
	o Overhead	170
Reasonableness test	o 14 tests at 550 ops/test each test run once every 20 sec	
	o Overhead	15
Statistical tests	o Iterative whiteness tests	300
	o Full whiteness	0
	o Covariance signatures	0
	o Sample mean	700
	o Pattern recognition	0
Data recording	Set up asynchronous transmission	50
	Determine device code	
	Determine new mode code	
	Assume 1 mode change per 30 seconds	

TABLE C3. BUS LOADING (FAULT-FREE ENVIRONMENT, NO PILOT INTERACTION)

Function	Description	WPS
Bit data collection	o 3 words of limit data for 69 LRU's	207
	o 1 status word for each bus address	20
	o Overhead	25
Reasonableness tests	o Collect test results from fusion processors 50 tests at 20 words per 20 sec	50
Statistical tests	o Iterative whiteness test	40
	o Full whiteness test (data storage only)	50
	o Covariance signatures (same data as above)	0
	o Sample mean (same data as above)	0
	o Pattern recognition	0
Data recording	o Mode change record to SMM contains: o Time 1 wd o Device/function 10 1 wd o New mode 1 wd o Overhead 12 wd Assume mode change every 30 sec	5

TABLE C4. SYSTEM MASS MEMORY STORAGE (FAULT-FREE ENVIRONMENT)
(Does not include computer load modules or fault data recording)

Function	Description	Size
BIT data collection	None	0
Reasonableness tests	None	0
Statistical tests	o Iterative whiteness tests o Full whiteness o Covariance signatures o Sample mean o Pattern recognition	5,200
Data recording	Change of LRU operating mode or change of function mode o Time 1 o Function/LRU 1 words per change o New mode 1 o Data tag 1	
	Initial state for 70 LRU's and 11 functions	352
	Plus 500 changes during mission	2,000
	Overhead	200

TABLE C4. SYSTEM MASS MEMORY STORAGE (FAULT-FREE ENVIRONMENT)
(Does not include computer load modules or fault data recording) (Continued)

Function	Description	Size
Self test initialization	Record of request and results	
	o Time	1
	o Subsystem/LRU	1
	o Results	6
Fault verification (active diagnostics)	None	-
Fault isolation	Fault free	2,800
Fault isolation	(Algorithms operate on data described in system status maintenance section)	-
Failure occurrence threshold testing	o 3 thresholds for each test and for each LRU (515 tests, 70 LRU's) o Reconfiguration scheme code for each type of failure	600

TABLE C4. SYSTEM MASS MEMORY STORAGE (FAULT-FREE ENVIRONMENT) (CONTINUED)

Function	Description	Size
Status request	None	0
Suspicion identification	Index of tests for each suspicion type (assume 50 suspicions and 512 tests)	1,600
	Table of test descriptions for pilot	
	General description indexes 1 per test	512
	Specific test description 4 per test	2,048
	Generic table: 15 groups at 5 words each	75
System status maintenance	Test failure status	
	Status of 52 tests 7 words of data including:	
	o No. times failed	
	o No. times failed since fault tolerance	
	o Current failure rate	
	o Maximum failure rate	
	o Time of maximum failure rate	
	Test threshold table (512 by 6)	3,072
	Fault snapshot file (4 words per fault)	4
	LRU status table	840
	Subsystem versus LRU table (70 boxes versus 256 modes) Includes graceful degradation schemes	1,120

TABLE C5. THROUGHPUT (OPERATIONS NECESSARY FOR PILOT INTERACTION AND FAULT ANALYSIS)

Function	Description	Operations
Statistical tests	Full whiteness	2,000
	Covariance signature	2,000
	Pattern recognition	1,000
Self-test initialization	Decode command	100
	Make self-test OK decision	20
	Initiate test	20
	Send results to display	20
	Set up for fault tolerance	40
Status request	Decode command	130
	Format status data	170
Suspicion identification	LRU decode command	50
	Access SMM data	15
	Fill data buffer for C&D	120
	Initiate transmission	5
	Set up fault tolerance	40
System status maintenaznce	Calculate data addresses	75
	Request data	35
	Update parameters	200
	Store in mass memory	90
Data recording	Acquire fault data	250
	Set up transfer to mass memory	50
Fault verification	Initiate failed test	70
	Check results for repeat of error	30
Fault isolation (active diagnostics)	Perform 50 branch decisions on fault tree	3,000
	o Acquire branch information 20	
	o Get data to make decision 25	
	o Make decision 15	

TABLE C5. THROUGHPUT (OPERATIONS NECESSARY FOR PILOT INTERACTION AND FAULT ANALYSIS) (CONTINUED)

Function	Description	Operations
Fault isolation (correlation of fault records)	Correlation schemes include:	
	o Logical analysis	400
	o Acquisition of historical data	200
	o Test of thresholds	400
Failure occurrence threshold testing	Acquire three threshold values and test against newly computed failure statistics	75
Emergency (time critical) handling procedures	Compare test-fail to emergency index table	30
Fault status	Access failed subsystem-LRU code and place in fault tolerance interface buffer	100
Reconfiguration verification	Identify and rerun failure-indicating test	200

TABLE C6. BUS LOADING (WORDS NECESSARY FOR NONMONITORING FUNCTIONS)

Function	Description	Words
Statistical tests	Retrieve data from mass memory	1,500
Self-test initialization	Coded initialization command from C&D	2
	Initiate self-test command to box/subsystem	2
	Results from box/subsystem to mission comp	10
	Formatted results to display	10
	Overhead	26
Status request	C&D request interface	50
	Status data send	80
	Overhead	120
Suspicion identification	Suspicion code to mission processor	5
	16 words of test pointers from mass memory	16
	Test results to C&D	60
	Overhead	19
System status maintenance	Request and reception of history data from mass	150
	Memory transmission of updated records	150
Data recording	Fault data to mass memory	65
	Fault data acquisition from subsystems	35
Fault verification	Send command to subsystem to restart test	10
	Receive data from subsystem	20
Fault isolation (active diagnostics)	50 mass memory data acquisitions	1,000
Fault isolation (correlation of flight records)	Acquisition of historical data	1,000
Failure occurrence Threshold testing	Three thresholds from mass memory	20

TABLE C6. BUS LOADING (WORDS NECESSARY FOR NONMONITORING FUNCTIONS) (CONTINUED)

Function	Description	Words
Emergency (time critical) handling procedures	None	-
Fault status	Transfer data to fault tolerance subsystem	50
Reconfiguration verification	Words to direct rerun of failure-indicating tests and to receive results	50

TABLE C7. SYSTEM MASS MEMORY STORAGE
(per Fault or Pilot Directive)

Function	Description	Words
Data recording	40 words per fault plus 5-word tag	45
Emergency (time critical) handling procedures	Log time, test, and operational mode	30
Fault status	Log reconfiguration status word, time	30
Reconfiguration verification	Log result notification from fault tolerance	20

APPENDIX D

REASONABLENESS TEST LOCATION

Preflight

Area	Test	Location of test
ADAM	Poll ADAM for indication of airfield and flat ground at NAV derived position	Mission processor
Air data	EDA test on total temperature	Mission processor
	Limit test of total temperature reading between -40°C and 120°C	Mission processor
	EDA test on total pressure minus static pressure	Mission processor
AHRS	Voting tests on AHRS and INS sensors measurement of pitch	Mission processor
	EDA test performed on difference between AHRS computed bearing and INS bearing	Mission processor
Guidance	Pilot visual validation of manual control surface exercise	Mission processor
	Pilot visual validation of automatic control surface exercise	Mission processor
ICNIA–GPS	Limit test on difference in position derived from GPS and table lookup of local TACAN beacon	ICNIA
ICNIA–TACAN	Limit test on difference in position derived from GPS and table lookup of local TACAN beacon	ICNIA
INS	EDA test on position from INS	Mission processor
	EDA test on velocity from INS	Mission processor
	Voting tests on AHRS and INS sensors measurement of pitch	Mission processor
	EDA test performed on difference between AHRS computed bearing and INS bearing	Mission processor
Stores	Display ordnance configuration to pilot for confirmation	Mission processor

Reasonableness Test Location (Continued)

In-flight

Area	Test	Location of test
Air data	EDA test on static pressure sensor	Mission processor
	EDA test on total temperature sensor	Mission processor
	EDA test on pitot pressure when in cruise mode	Mission processor
	Voting test on pitch measurements from AOA sensors AHRS, and INS	Mission processor
Guidance	EDA test on cross track error	Mission processor
	EDA test on course error	Mission processor
	ROC test on flight path angle	Mission processor
	ROC test on flight path pitch commands	Mission processor
	Limit test on flight path angle	Mission processor
	Limit test on flight path pitch commands	Mission processor
AHRS	ROC test on sensor when navigation is in single (or multiple-unintegrated) sensor mode (attitude)	Mission processor
	EDA test on sensor when navigation is in single (or multiple-unintegrated) sensor mode (attitude)	Mission processor
	Voting test on pitch measurements from AOA sensors AHRS, and INS	Mission processor
	Limit test on difference of AHRS and INS bearing measurements	Mission processor
ICNIA–GPS	[illegible] tests on position estimates from [illegible]S, TACAN and INS	Mission processor
	[illegible] on sensors when navigation is in single (or multiple-unintegrated sensor mode position)	Mission processor

Reasonableness Test Location (Continued)

In-flight

Area	Test	Location of test
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (velocity)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (position)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (velocity)	Mission processor
ICNIA—ILS	ROC test on glide slope deviation	ICNIA
	ROC test on ILS localizer bearing	ICNIA
	ROC test on ILS localizer deviation	ICNIA
ICNIA—JTIDS	Voting tests on position estimates from GPS, JTIDS, TACAN and INS	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (position)	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (velocity)	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (attitude)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (position)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (velocity)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (attitude)	Mission processor

Reasonableness Test Location (Continued)

In-flight

Area	Test	Location of test
	ROC test on JTIDS time of arrival	ICNIA
	ROC test on JTIDS emitter location signal	ICNIA
ICNIA–TACAN	Voting tests on position estimates from GPS, JTIDS, TACAN and INS	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (slant range)	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (bearing)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (slant range)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (bearing)	Mission processor
IIR Maverick	ROC testing on image processing tracks of target range	Image fusion
	ROC testing on image processing tracks of target bearing	Image fusion
INS	Voting tests on position estimates from GPS, JTIDS, TACAN and INS	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (attitude)	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (position)	Mission processor
	ROC test on sensors when navigation is in single (or multiple-unintegrated sensor mode (acceleration)	Mission processor

Reasonableness Test Location (Continued)

In-flight

Area	Test	Location of test
	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (velocity)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (attitude)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (position)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (acceleration)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (velocity)	Mission processor
	Voting test on pitch measurements from AOA sensors, AHRS, and INS	Mission processor
	Limit test on difference of AHRS and INS bearing measurements	Mission processor
Laser desig.	ROC tests on aiming angles for laser designator	Target fusion
Laser det/tr	ROC tests on aiming angles for laser detection/tracker	Target fusion
	Limit test on detection angle for detector/tracker. (Is detection angle within window of designator sweep? Limits are defined by window of designator sweep.)	Target fusion
Laser ill/rang	ROC tests on range to target from laser illuminator/ranger	Target fusion
MMW	ROC testing on image processing tracks of target range	Image fusion
	ROC testing on image processing tracks of target range	Image fusion

Reasonableness Test Location (Continued)

In-flight

Area	Test	Location of test
Power cont.	Limit test on power load shift from Advanced Power Management System (APMS)	Mission processor
Radar alt.	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (altitude)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (altitude)	Mission processor
SAR	ROC testing on image processing tracks of target range	Image fusion
	ROC testing on image processing tracks of target bearing	Image fusion
Stores	Poll weapon status after firing to assure correct and complete release	Mission processor
Target proces.	ROC tests applied to absolute position calculated by target fusion processor	Mission processor
	ROC tests applied to absolute speed calculated by target fusion processor	Target fusion
	ROC tests applied to absolute heading calculated by target fusion processor	Target fusion
TF/TA	ROC test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (altitude)	Mission processor
	EDA test on sensors when navigation is in single (or multiple-unintegrated) sensor mode (altitude)	Mission processor
Threat fusion	ROC test on threat locations (failure threshold depends on class of threat)	Threat fusion
Weapon delivery processor	Monitor select, arm, and fuse commands for each weapon to ensure commands are in correct sequence	Stores processor

Reasonableness Test Location (Continued)

In-flight

Area	Test	Location of test
	EDA test on guidance error	Mission processor
	EDA test on revised guidance error	Mission processor
	Limit test on guidance error	Mission processor
	Limit test on revised guidance error	Mission processor

APPENDIX E
ESTIMATES OF CND AND RTOK RATES

"This level-of-success figure is derived from field maintenance data showing that 40% of the avionics equipment removed from an aircraft is fault-free and fully capable of satisfying its assigned mission function. This inability to identify malfunctioning equipment without ambiguity results in a 67% workload increase at organizational and intermediate maintenance levels."—Top Down Built-In Test Architecture Study LR29523, Lockheed-California Company, April 1980.(21)

"... typically 30% of inflight BIT-indicated faults could not be duplicated on the ground and typically 20% to 30% of the units which were faulted by BIT were found to be fault-free in the shop."—BIT False Alarms: An Important Factor in Operational Readiness, 1982 Proc. Annual Reliability and Maintainability Symposium, John Malcolm.(22)

"Cannot-duplicate events were found to be generally high and definitely impacting aircraft availability. The following presents a summary of the number of LRU-WUC's exceeding 10% CND rates for several typical aircraft."

Aircraft	CND rate (%)	Number of LRU-WUC's
A	10 to 24	18
B	10 to 48	22
C	10 to 57	11
D	10 to 50	18
E	16 to 43	5

"Bench-check-OK (BCO) events were also found high; examples of the ranges are:" (Note: BCO is equivalent to RTOK)

Aircraft	BCO rate (%)	Number of LRU-WUC's
A	10 to 46	15
B	10 to 44	24
C	11 to 50	13
D	10 to 75	27
E	14 to 61	5

· Modular Automatic Test Equipment Guides Sperry, June 1981.(23)

"Labor expended on CND's for eight typical subsystems ranged from 4 percent to 22 percent."—Design-for-Repair Concept Definition, AFAL-TR-79-1130, Hughes Aircraft Company, August 1979.(24)

The following are from presentations at Built-in-Test Equipment Requirements Workshop, Institute for Defense Analysis, Paper P-1600, August 1981.(3)

"In addition, experience shows that 20 to 40 percent of the items which were replaced because of a failure indication by BIT are later found to have no failure (principally based on data from both military and civilian aircraft maintenance experience)."—Executive Summary.

"...BIT false alarm rate between 20 and 30 percent in RADC studies"—BIT Programs, George Neumann, NAVMAT-04T.

"Other RADC studies involving nine different Air Force systems at numerous bases have shown unnecessary removal rates on the order of 40 percent with some systems as high as 89 percent."—BIT Programs, George Neumann, NAVMAT-04T.

"Airlines find that far less than 50 percent of boxes removed contained verified failures, especially auto pilots (the worst) which run 85 to 90 percent nonverified."—BIT Programs, George Neumann, NAVMAT-04T.

CND rate of 34% for ALQ-126—AN/ALQ-126B, Designing and Validating BIT, Ken Wilson, Maintenance Technology, Inc.

"A 30 to 40 percent RTOK rate was found quite universal"—BIT Specification and Demonstration Techniques, Capt. Dan Gleason, RADC.

"The CND rate is approximately 30 percent in military, in industry, and the airlines...", BIT Workshop Panel #3 Report.

Appendix F

EXISTING EXPERT SYSTEM APPLICATIONS TO TESTING AND MAINTENANCE

IDT: An Intelligent Diagnostic Tool (26)

Digital Equipment Corporation
Knowledge Engineering Group
Hal Shubin

Summary

IDT is an intelligent hardware diagnostic tool used to identify faults in PDP 11/03 computers. It selects and executes tests, and interprets the results. IDT is also able to modify its test selection strategy on the basis of results of previous tests as well as user-introduced opinions.

Physical Configuration and Operation

Two computers, besides the unit under test, make up the testing configuration. One of the computers is remote while the other is local to the unit under test. The remote computer, a VAX 11/780, contains the knowledge base, reasoning mechanisms and testing strategies. The local computer, a PDP 11/03, contains the diagnostic test series and the display software. The two computers communicate over a 4,800-baud telephone line.

The user initiates a diagnostic session by powering up the 11/03, which then telephones the 11/780 and logs in to a special account, initiates the user's display, and then retires to a passive role. Future activities of the diagnostic process are controlled by the 11/780 with the 11/03 passing messages, loading and running the tests when told to do so, and managing the display.

Formulation of the Diagnostic Process

IDT performs two basic functions:

a. It analyzes the results of the tests to determine which Field Replaceable Unit should be replaced.

b. It selects diagnostic tests from a set of tests. Test selection is based on the knowledge acquired from previous tests and from opinions entered by the user.

To accomplish the above functions IDT has—

a. A method for interpreting the results of the diagnostic tests (an adequate model of the unit under test must be developed).

b. A method for reasoning about interpreted test results.

c. A strategy for selecting the testing order.

REACTOR: An Expert System for Diagnosis and Treatment of Nuclear Reactor Accidents (27)

EG&G Idaho, Inc.

Summary

REACTOR is an expert system under development at EG&G Idaho, Inc., that will assist operators in the diagnosis and treatment of nuclear reactor accidents. As such, the nature of the decision process (online process control) is different from the usual expert system applications. A unique feature of REACTOR is the integration of event-oriented and function-oriented diagnostic strategies providing a useful combination for handling emergency situations.

Operation

The purpose of REACTOR is to monitor a nuclear reactor facility, detect deviations from normal operating conditions, determine the significance of the situation, and recommend an appropriate response. It performs these tasks by operating on a large knowledge base with a procedure that reasons both forward and backward. The reasoning process was adapted from Winston and Horn's animal identification system (28). The system reasons forward from known facts until a conclusion can be reached. If not enough information is available to reach a conclusion, the system reasons backward to determine what information it needs to know. REACTOR will then query the plant instruments or the operator to fill the gaps in its knowledge.

Knowledge Base

REACTOR's knowledge base contains two types of knowledge: function-oriented knowledge and event-oriented knowledge. The former concerns the configuration of the reactor system and how its components work together to perform its activity; the latter describes the expected behavior of the reactor under known accident conditions. Event-oriented knowledge has been gathered from past experience with actual accidents, experiments in test reactors, and analysis of computer simulation models.

Function-oriented information is considered when an event does not match an expected pattern of preanalyzed events. All other times, event-oriented knowledge is used that is contained in a series of "if-then" rules.

The function-oriented capabilities of REACTOR are handled by the response tree technique. A response tree is a diagram that shows the success paths that can be used to provide a given function. In REACTOR the function is safety.

A CSA Model-Based Nuclear Power Plant Consultant (29)

W. E. Underwood
School of Information and Computer Science
Georgia Institute of Technology

Summary

This system is an experimental computer-based nuclear power plant consultant. The interference procedures interpret observations of a particular plant situation in terms of a commonsense algorithm (CSA) network that characterizes the normal and abormal events of a pressurized water reactor (PWR) plant. This effort, undertaken in the academic community, takes a conventional problem—developing a diagnostic capability for a known physical system—and approaches its solution through an unconventional means—the merging of commonsense algorithms and expert systems technology. The researchers investigated the use of CSA's and expert systems technology in representing knowledge of nuclear power plants for use in problem diagnosis and intervention.

System Description

The current system consists of a 350-event CSA model of a PWR coolant system constructed by a nuclear engineering expert, a CSA network simulator for designing and testing the models; and a diagnostic program that uses a forward chaining control strategy. The CSA network currently consists of models of several systems within the plant. In addition, events that are symptomatic of system problems and that cause alarms and automatic control actions are represented. Diagnostic rules are also represented in the CSA network. Of the various expert systems, this prototype most closely resembles CASNET or EXPERT. The prototype provides a knowledge base of nuclear power plant operation, procedures, and experience coupled with an automatic diagnostic capability.

Commonsense Algorithm Network Models

The CSA representation for physical mechanisms consists of four event-types and nine relations. The four events are actions, tendencies, states, and stage changes. The nine relations are one-shot causality, continuous causality, repetitive causality, state coupling, equivalence, anatagonism, enablement, threshold, and rate confluence. Events and relations particular to the subject model are developed with expert assistance.

Control Strategy

A consultation usually begins with an operator requesting diagnosis of the cause of some abnormal event. The control strategy first indexes into the CSA net to locate these events. A forward chaining control stragegy is used. When a causal event is inferred that has immediate effects that are observable but not verified, the controller asks the operator to verify these in order to further confirm the inference. The CSA network is also able to interpret the meaning of observations that are seemingly contradictory and thus resolve many of the apparent conflicts.

DART: Diagnostic Assistance Reference Tool (30)

Michael R. Genesereth
Stanford University

Summary

DART is a test generation algorithm that was developed as an automated diagnostician for the diagnosis of computer hardware faults. The algorithm uses a general inference procedure to compute suspect components and generate discriminatory tests from information about the design of the device being diagnosed. The program accepts a statement of a system malfunction in a formal language, suggests tests, accepts the results, and ultimately pinpoints the components responsible for the failure. The DART algorithm differs from the currently available medical diagnosis expert systems. These systems all use rules that associate symptoms with possible diseases. The DART program contains no information about how computers fail. Instead, it works directly from information about intended structure (a machine's parts and their interconnections) and expected behavior (equations, rules, or procedures that relate inputs and outputs).

Operational Concept

While the cost of executing a single test is usually small and the number of tests needed to pinpoint a fault is, at worst, linear in the number of components, the cost of generating appropriate tests grows polynomially or exponentially. The DART program meets this difficulty by exploiting the hierarchy inherent in most computer system designs. The program first diagnoses the system at a high level of abstraction to determine the major subcomponent in which the fault lies. It then focuses its attention on the next lower level, and repeats this progression until it can identify a replaceable part. In this way, the number of components under consideration at any one time is kept small, and the cost of test generation remains manageable.

Within each level, DART uses a deductive procedure to compute suspects and generate tests. All symptoms are expressed as violations of expected behavior. Starting with a symptom of this type, DART reasons backwards from the expected behavior to discover why it was expected, and in so doing, produces a justification for its conclusion. The next step is to discriminate among these suspects. DART starts with a behavioral rule for one of the suspects and works forward to observable outputs and backwards to modifiable inputs.

The DART Algorithm

The DART procedure begins with the design description for the device under test and a set of observed symptoms. It produces as output the minimal set of replaceable parts that will correct the error. Especially useful to the procedure are the assumptions that the fault occurs within a single replaceable part and is not intermittent. The workhorse of the DART algorithm is a general inference procedure. In the current implementation, this procedure is linear-input resolution, guided by a set of explicit metalevel control rules.

Physical Configuration

The DART algorithm implementation was done in COMMON LISP with the help of a data base and inference system called MRS. In all cases the algorithm was able to generate appropriate tests and diagnose the underlying faults. The time required to diagnose each case on a VAX 11/780 was on the order of minutes.

ACRONYMS

ACP	Advisory Control Panel
ADAM	Advanced Digital Avionics Map
ADC	air data computer
ADF	automatic direction finding
AETMS	Airborne Electronic Terrain Mapping System
AF	Air Force
AFFDL	Air Force Flight Dynamics Laboratory
AFWAL	Air Force Wright Aeronautical Laboratory
AGM	air-to-ground missile
AHRS	Attitude, Heading Reference System
AI	artificial intelligence
ALCM	Air Launched Cruise Missile
AOA	angle of attack
AP	armament panel
APMS	Advanced Power Management System
ASAT	Antisattelite
ASID	Advanced System Integration Demonstration
ASP	Avionics Status Panel
ATE	automatic test equipment
BCI	Bus Control Interface
BCM	Bus Control Module
BCO	bench check ok
BCP	BIT Control Panel
BIT	built-in test
BITE	built-in test equipment
BITR	built-in test register
BIU	Bus Interface Unit
BMAC	Boeing Military Airplane Company
CCIP	continuously computed impact point
CFE	contractor-furnished equipment
CI	configuration item
CND	cannot duplicate
DAIS	Digital Avionics Information System
DEK	Data Entry Keyboard

DSMU	Display Switch and Memory Unit
DTU	Data Transfer Unit
EDA	excessive deviation from average
EMI	electromagnetic interference
EROM	erasable read-only memory
EW	electronic warfare
FCC	Fire Control Computer
FCNP	Fire Control Navigation Panel
FIT	Fault Isolation Test
FLIR	forward looking infrared
FOM	figure of merit
GPS	Global Positioning System
HARS	Heading and Attitude Reference System
HSD	Horizontal Situation Display
ICAI	Intelligent Computer-Aided Instruction
ICNIA	Integrated Communication, Navigation, and Identification Avionics
IFF	Identification Friend or Foe
IFFC	integrated fire and flight control
IFTP	inflight test program
ILS	Instrument Landing System
IM	interface module
IMFK	Integrated Multifunction Keyboard
INS	Inertial Navigation System
IR	infrared
ITM	Integrated Testing and Maintenance
IUS	Inertial Upper Stage
JTIDS	Joint Tactical Information Distribution System
LND	land
LRU	line replaceable unit
MAADS	Multibus Avionic Architecture Design Study
MCL	Master Caution Lamp
MFK	Multifunction Keyboard
MFL	maintenance fault list
MIU	Maintenance Interface Unit
MMD	Master Monitor Display
MMP	Master Monitor Panel

MMU	Mass Memory Unit
MMW	millimeter wave
MPD	Multipurpose Display
MPDG	Multipurpose Display Generator
MTBF	mean time between failures
MTU	Multiplex Terminal Unit
MUX	multiplex
NABIT	nonavionic built-in test
NAV	navigation
NTWS	New Threat Warning System
OFP	Operational Flight Program
OPS	operations per second
OTP	Operational Test Program
PCP	Processor Control Panel
PFL	pilot fault list
PMI	Performance Monitoring Interface
R&D	research and development
ROC	rate of change
RS	running sum
RT	Remote Terminal
RTOK	retest OK
SAR	synthetic aperture radar
SCU	Sensor Control Unit
SMM	system mass memory
SRAM	Short Range Attack Missile
SRU	shop replaceable unit
ST	self-test
TACAN	Tactical Area Navigation
TCU	Terminal Control Unit
TERCOM	Terrain Contour Matching
TF/TA	Terrain Following/Terrain Avoidance
TOF	takeoff
VLSI	very large scale integration
VSD	Vertical Situation Display
WPS	words per second

☆ U.S. GOVERNMENT PRINTING OFFICE 1984—7:9-062/6 65